University of Pennsylvania
The Moore School of Electrical Engineering
200 South 33rd Street
Philadelphia, PA 19104-6390

OPTICAL COMPUTING BASED ON NEURONAL MODELS

Final Report

Prepared by

Nabil H. Farhat

For
Naval Research Laboratory
4555 Overlook Avenue, S.W.
Washington, D.C. 20375
Attn: Code 5709

Under

Grant No. N00014-85-K-2036

May 1988

EO/MO Report No. 14
Period: 9/15/85 - 2/15/88

University of Pennsylvania
The Moore School of Electrical Engineering
200 South 33rd Street
Philadelphia, PA 19104-6390

OPTICAL COMPUTING BASED ON NEURONAL MODELS

Final Report

Prepared by

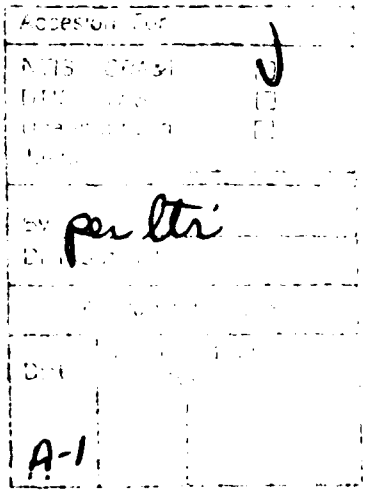
Nabil H. Farhat

For
Naval Research Laboratory
4555 Overlook Avenue, S.W.
Washington, D.C. 20375
Attn: Code 5709

Under

Grant No. N00014-85-K-2036

May 1988



EO/MO Report No. 14
Period: 9/15/85 - 2/15/88

TABLE OF CONTENTS

REPORT DISTRIBUTION LIST

	Copies
Naval Research Laboratory 4555 Overlook Ave., S.W. Washington, D.C. 20375 Attn: Code 5709	1
ONRR National Academy of Sciences 2100 Pennsylvania Ave., N.W. Washington, D.C. 20037-3202	1
Director Naval Research Laboratory 3555 Overlook Ave., S.W. Washington, D.C. 20375 Attn: Code 2627	6
Defense Technical Information Center Bldg. 5, Cameron Station Alexandria, VA 22314	12
Program Management Office 1400 Wilson Blvd. Arlington, VA 22209-2308 Attn: Dr. John Neff	1

OPTICAL COMPUTING BASED ON NEURONAL MODELS

1. INTRODUCTION

The ultimate goal of the research work carried out under this grant is understanding the computational algorithms used by the nervous system and development of systems that emulate, match, or surpass in their performance the computational power of biological brain. Tasks such as seeing, hearing, touch, walking, and cognition are far too complex for existing sequential digital computers. Therefore new architectures, hardware, and algorithms modeled after neural circuits must be considered in order to deal with real-world problems.

Neural net models and their analogs represent a new approach to collective signal processing that is robust, fault tolerant and can be extremely fast. These properties stem directly from the massive interconnectivity of neurons (the logic elements) in the brain and their ability to perform many-to-one mappings with varied degree of nonlinearity and to store information as weights of the links between them, i.e., their synaptic interconnections, in a distributed non-localized manner. As a result signal processing tasks such as nearest neighbor searches in associative memory can be performed in time durations equal to a few time constants of the decision making elements, the neurons, of the net. We note that the switching time-constant of a biological neuron is of the order of a few milliseconds. Artificial neurons (electronic or opto-electronic decision making elements) can be made to be a thousand to a million times faster. Artificial neural nets can therefore be expected to function for example as content addressable associative memory or to perform complex computational tasks such as combinatorial optimization and minimization

which are encountered in self-organization and learning (self programming), computational vision, imaging, inverse scattering, super-resolution and automated recognition from partial, (sketchy) information, extremely fast in a time scale that exceeds by far the capability of even the most powerful present day serial computer. For these reasons electronic and opto-electronic analogs and implementations of neural nets are attracting today considerable attention. The optics in the opto-electronic implementations provide the needed parallelism and massive interconnectivity while the decision making elements are realized electronically heralding an ultimate marriage of VLSI and optics. It should be kept in mind however that research and advances in optical bistability devices (OBDs) and nonlinear optics and optical materials, promise to furnish also all-optical decision making elements and eventually neural nets in which both the interconnections and decision making are performed optically with the electronics being used only for control and assessment of the state of the net. The combination of optics and electronics and the potential for exploiting advances in opto-electronic components and materials (for example; nonvolatile spatial light modulators for realizing programmable synaptic or interconnectivity masks (plasticity) and OBDs for decision making) promise also that embodiments of neural nets can be compact and have low power consumption. Such embodiments, being primarily analog, are leading to rekindling of interest in analog computation whose development has been curtailed by the explosive progress in digital computing.

In associative memory applications, the strength of interconnection between the "neurons" of the net is determined by the entities one wishes to store in the net. Usually these entities need to be in the form of uncorrelated binary representations of the original data. Specific storage

"recipes" based on a Hebbian model of learning (outer-product storage algorithm), or variations thereof, are employed then to first calculate the connectivity matrix then set the weights of links between neurons accordingly. In this sense the memory is explicitly programmed i.e. taught what it should know and should be cognizant of. This mode of programming a net is sometimes called **hard learning.** What is most intriguing however, is that neural nets can also be made to be self-organizing and learning i.e., to become self-programming **(soft learning)** through a process of automated connectivity weight modification driven by the entities presented to them for learning. This alleviates one of the major constraints of neural nets: programming complexity, and makes them a much more attractive and powerful tool for **neuromorphic signal and knowledge processing.** The combination of neural nets, **Boltzmann machines,** and **simulated annealing** concepts with high speed opto-electronic implementations promise, as demonstrated by research carried out under this grant ([1]-[2] and this report), to produce high-speed artificial neural net processors with stochastic rules for decision making and state update that can form their own internal representations (connectivity weights) of outside world data they are presented with, regardless whether the data is correlated or not, in a manner very analogous to the way the brain is believed to form its own symbolic representation of reality. This is an exciting prospect and has far reaching implications for smart sensing and recognition, and artificial intelligence as a whole. The use of noise in **stochastic learning** can shed light on the way nature has managed to turn noise present in biological neural nets to work to its advantage and makes stochastic learning, as opposed to deterministic learning schemes more biologically plausible. Such learning is probabilistic in nature aimed at capturing the probability density function

of the environmental representation the net is exposed to. Probabilistic learning is therefore naturally compatible with real environmental representations that are fuzzy in nature. Exploratory work at the University of Pennsylvania is showing that optics can play an important role in the implementation and speeding up of adaptive learning algorithm, such as the simulated annealing and the error back-propagation algorithms, in such self-organizing nets and can lead to their use in automated robust recognition of entities the nets have had a chance to learn earlier either with or without the aid of a teacher (supervised or unsupervised learning) by repeated exposure to them when the net is in its learning mode. One can envision modules of such self-teaching neural nets trained to recognize and create symbols of certain features found in natural scenes, patterns or other input signals. Such modules could be used collectively for higher level processing where their output symbols are fused to form better or more reliable interpretation or assessment of the environmental input. The implication of this for autonomous systems are obvious but the achievement of such scenarios requires further concerted research.

Learning in neural nets is not rote but involves generalization, i.e. the net can recognize an input as a member of a class of entities it became familiar with earlier even though that specific input was not specifically among those shown to it earlier. This property can be extremely useful for accelerating teaching sessions in that one need not think of and present to the learning net all possible associations it is supposed to recognize in order to make it useful. This property relegates however a degree of decision making to the net perhaps beyond what we are ordinarily accustomed to in signal processing systems. Thorough understanding of learning processes and how a network generalizes is therefore desired to alleviate

apprehensions and uncertainties stemming from the inclusion of "thinking networks" in man made systems that share with him the decision making process. Such understanding can be realized only through insights gained normally by theoretical analysis and with software and hardware simulation tools. Being highly nonlinear, neural nets (as for example in higher cortical areas), and their models are often difficult to analyze. Numerical simulation of neural nets, even relatively small multilayered self-organizing nets, are proving to be computationally too intensive and therefore unacceptably time consuming which is hindering progress in the field. It is for this reason that analog systems in which neural net behavior can be modeled and studied dynamically at speeds that can be several orders of magnitude faster than in numerical simulation are an important component of our ongoing studies and the future research directions stemming from it. This work is pointing towards neural nets as nonlinear dynamical systems that are characterized by their phase space behavior and concepts of attractors, chaos and fractal dimensions. This will in our opinion provide an infusion of powerful concepts of nonlinearity, collective behavior, and iterative processing into optical processing and artificial neurodynamical systems.

Another intriguing promise of neural nets is their ability to store and retrieve information in a sequential or cyclic manner where a chain of entities can be stored and recalled in a hetero-associative sequential or cyclic fashion. This can provide a crude but simple way for forming, shaping, and controlling the limit cycle (trajectory) of a neural net in its phase-space. This property together with that of generalization, mentioned earlier, are important for work in pattern recognition in general and are being intensively studied at our Electro-Optics and Microwave Optics

Laboratory in the context of distortionless radar target recognition as described in earlier work (see references listed in this report and in [1] and [2]. The results of this work are expected to be general and would be beneficial to active and passive machine vision*.

Being highly nonlinear neural nets possess complex, rich phase-space behavior that exhibits or is in principle capable of exhibiting the following general features of nonlinear systems:

- o Fixed points or limit points in phase-space that act as attractors with prescribed basins of attraction that are formed explicitly (hard learning) or implicitly (soft learning) by Hebbian based rules.
- o Fixed limit cycles or closed trajectories in phase-space that act also as attractors.
- o Fixed open trajectories that act as attractors.
- o Modification and control of fixed limit points, limit cycles and open trajectories by external and/or contextual input or by adaptive thresholding.
- o Bifurcation and chaotic behavior.

Neuromorphic signal and knowledge processing systems (whether optical, electronic, or opto-electronic) must be able to draw upon and make use of these features to achieve powerful signal processing functions. Such functions include:

*Machines that utilize active illumination to discern and perceive the environment or utilize natural scene illumination or emission for the same purpose.

- o Nearest neighbor searches
- o Combinatorial optimization by minimization of cost-functions
- o Solution of ill-posed problems of the kind encountered in vision, remote sensing, and inverse scattering
- o Feature extraction: self-organization, learning and self-programing
- o Generalization
- o Sequential and cyclic retention and recall
- o Higher order or more complex computations in phase-space. (e.g. spoken language processing)

All the above are issues that provide motivation to our neural net and opto-electronic implementation research, both current and future.

The ultimate realization of neuromorphic systems for wide use in signal processing applications is not a trivial task. It requires vigorous research and development in three primary areas: neuroscience - to increase our understanding of the anatomical, physiological, and biochemical properties, and function of neural tissue (neural nets) in order to identify those attributes that might help in their modeling and that can be usefully applied in artificial systems; the study of opto-electronic architectures and implementations, and vigorous device development based on advances in linear and nonlinear optical materials for efficient implementation of programmable synaptic weights (artificial plasticity) and sensitive optical decision making elements capable of performing at lower threshold then present day devices. Thus synergisim between a triad of research activities: neuroscience; mathematical modeling and analysis coupled with architectures, implementations, and programming; and material research is

called for. Our future research in neurodynamics will continue to be influenced by developments in these fields.

2. RESEARCH ACCOMPLISHMENTS

The first parts of our research program under this grant were concerned with the modeling, simulation, and implementation of fully interconnected neural networks and were reported upon in detail in two previous annual reports [1],[2]. Work with fully interconnected nets and specifically a comparison of inner product versus outer product schemes for associative storage and recall (see Appendix III) revealed to us that one of the most distinctive property of neural nets that is worth considering in our research efforts is self-organization and learning. Self-organization (adaptivity) and learning seems to be what sets neural net processing apart from other approaches to signal processing. Hence our efforts have since been more concerned with learning and self-programmability in neural nets. Learning requires layered nets in which one can clearly distinguish input, output, and hidden (buffer) groups of neurons with proscribed communication patterns among them. Such nets are hence non-fully interconnected. To this end we have devised a scheme for partitioning existing opto-electronic vector-matrix multiplier architectures into any number of desired layers (see Appendices II and III). Learning requires plasticity, i.e. modifiable weights of connections between neurons. In our work such plasticity is achieved primarily through the use of programmable nonvolatile spatial light modulators (SLMs) such as the magneto-optic SLM (MOSLM).* We have devised a

*Nonvolatile ferroelectric liquid crystal SLM can also be used. These however are not available commercially yet.

new scheme for driving a commercially available 48x48 element MOSLM at a frame refresh time of 1 msec demonstrating thereby that synaptic modification in a 24 binary neuron net constructed around this MOSLM can take place if desired in a time period as short as 1 msec.

From the outset we have concentrated our effort on stochastic rather than deterministic learning for several reasons. Stochastic or probabilistic learning is more compatible with the uncertainty of most environments in which learning neural nets are expected to operate. In addition operation and study of stochastic neural nets could shed some light on the way nature has harnessed noise present in biological neural nets to work to its advantage. Learning in stochastic neural nets involves finding the global minimum of an energy function associated with the network by introducing uncertainty in the state update rule of neurons in the net. Conventionally this is done by a simulated annealing algorithm in a context of a Boltzmann machine that was devised to be carried out on a serial computer and is frequently used in the solution of combinatorial optimization problems. Software implementation of the process however is time consuming. For example finding the optimal wiring layout for a typical IC chip might take 24 hours on mainframe computer. We have developed therefore a method for accelerating the annealing time in an opto-electronic stochastic neural net that can be several orders of magnitude faster than serial digital methods. As a result, stochastic learning in such nets can be speeded-up by the same factor. The method involves the use of noisy thresholding of the neurons in the net which introduces controlled shaking of the energy landscape of the net and prevents the net from getting trapped in a state of local energy minimum improving thereby its chances of finding the ground state (global minimum) or one close to it. By introducing the

noise in the net in bursts of decaying magnitude the chances of converging onto a low-lying energy state and staying in it are enhanced considerably. (Such controlled annealing profiles or annealing schedules are also useful in stochastic learning with binary weight to be described later.) The results of numerical and experimental simulations (see Appendix V) show that the noisy thresholding scheme is quite effective. We used the noisy prototype thresholding scheme in a network of 16 neurons with random bipolar binary weights implemented in opto-electronic hardware. The results show that the net can find the ground state in 35τ where τ is the time constant of the neurons in the net. This means that for a net with neurons of τ = 1μsec response time the net can be annealed in 35 μsec and this is independent of the number of neurons in the net as noise in our scheme is injected optically onto all neurons simultaneously by projecting a portion of the snow pattern appearing on an open channel T.V. receiver onto the photodetector array segment of the opto-electronic neural net.

The pixel transmittance of the MOSLM mentioned earlier is binary. Most known learning algorithms require small incremental changes in the connection strengths between neurons of the net. This means multivalued weights are necessary precluding the use of MOSLMs despite their highly desirable nonvolatile nature (storage capability). Small incremental changes in the weights is a requisite for convergence of the learning algorithm. To overcome this limitation we have devised a method for stochastic learning with binary weights. The method combines multiple time-constant annealing bursts and dead-zone limiting as detailed in Appendix VI.

Fast annealing by noisy thresholding and stochastic learning with binary weights are significant developments that enabled successful

operation recently in our work of the first bimodal stochastic optical learning machine as detailed (see Appendix VI). The machine consists of 24 unipolar binary neurons, 24x24 bipolar binary connectivity mask implemented in a 48x48 computer controlled MOSLM, and LED and photodetector arrays with associated thresholding amplifiers and LED drivers for the neurons themselves. Preliminary results of the learning capabilities show that the net can learn a set of 3 associations in a time interval ranging between 10 minutes to 60 minutes with relatively slow (60 msec) neurons. Preliminary results are shown in Fig. 1. Slow neurons were chosen deliberately to permit visual observation of the evolving state vector of the net as represented by the LED array of the net during the various stages of learning.

3. CONCLUSIONS

Research effort under this grant has led to the demonstration of the first stochastic opto-electronic learning machine employing fast annealing by noisy thresholding and stochastic learning with binary weights. The prototype machine of 24 neurons now operational in our laboratory provides a valuable vehicle for studying the dynamics of stochastic neural nets. As such, the net can be viewed as an opto-electronic analog computer that can perform iterative mappings, do stochastic searches of the energy landscape, self-organize and learn, and act as associative memory after learning is completed. We will continue our studies of this and larger versions of the machine (a subject for renewal proposal under preparation) in order to gain better understanding of the behavior of such machines as artificial neurodynamical systems and explore a host of intriguing applications involving solution of combinatorial optimization problems of the kind encountered in vision, remote sensing and inverse scattering.

Experimental Result – Learning

Number of input neurons : 8
Number of hidden neurons : 8
Number of output neurons : 8
There are 3 mappings required to learn:
10100000 - 10100000
00010010 - 00010010
00000101 - 00000101
Annealing time : 1 second
In one learning cycle, each mapping is presented to the net 3 times
One learning cycle time : 30 second
Results:

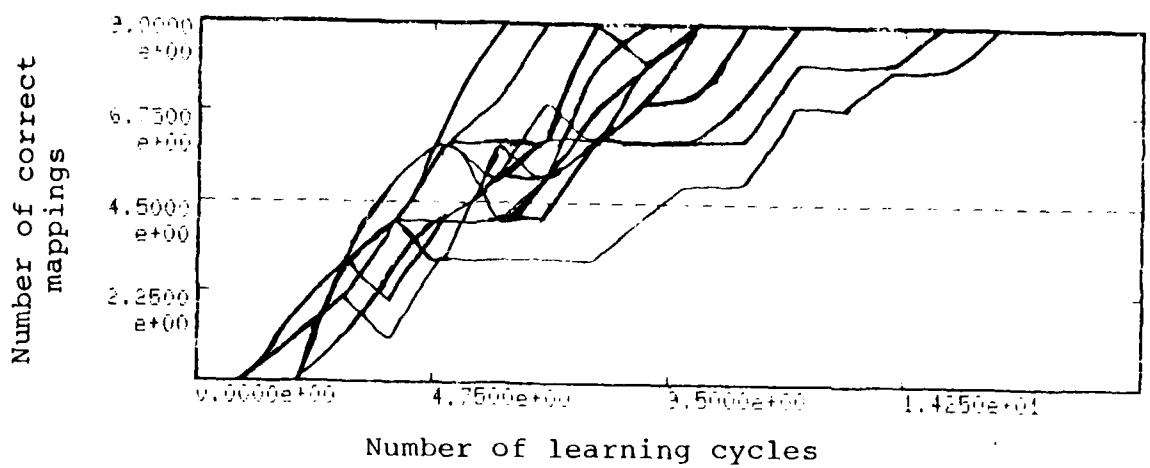


Fig. 1

4. REFERENCES

1. N. Farhat, "Optical Computing Based on the Hopfield Model for Neural Nets," University of Pennsylvania Report No. EO/MO 10, prepared for NRL, May 1986.

2. N. Farhat, "Optical Computing Based on Neuronal Models," University of Pennsylvania Report No. EO/MO 12, October 1987.

5. PUBLICATIONS AND OTHER ACTIVITIES

Following is a list of journal and conference proceeding publications describing research carried out under this grant:

1. D. Psaltis and N. Farhat, "A new approach to optical information processing based on the Hopfield Model," in Digest of the Thirteenth Congress of the International Commission on Optics, (ICO-13), Sapporo, Japan (1984), p. 24. (pre-grant publication).

2. D. Psaltis and N. Farhat, "Optical information processing based on an associative-memory model of neural nets with thresholding and feedback," Opt. Lett., 10, 98 (1985).

3. N.H. Farhat and D. Psaltis, A. Prata, and E. Paek, "Optical implementation of the Hopfield Model," Appl. Opt., 24, 1469 (1985), pp. 1469-1475.

4. N.H. Farhat and D. Psaltis, "Architectures for optical implementation of 2-D content addressable memories," in Technical Digest, Optical Society of America Annual Meeting (Optical Society of America, Washington, DC, 1985), paper WT3.

5. K.S. Lee and N.H. Farhat, "Content addressable memory with smooth transition and adaptive thresholding," in Technical Digest, Optical Society of America Annual Meeting (Optical Society of America, Washington, DC, 1985), paper WJ35.

6. N. Farhat, S. Miyahara and K.S. Lee, "Optical implementation of 2-D neural nets and their application in recognition of radar targets," in Neural Networks for Computing, J.S. Denker, Ed. (American Institute of Physics, New York), pp. 146-152.

7. N. Farhat and D. Psaltis, "Optical implementation of associative memory based on models of neural networks," in Optical Signal Processing, J.L. Horner, Ed. (Academic, New York, 1987), pp. 129-162.

8. N.H. Farhat, "Architectures for opto-electronic analogs of self-organizing neural networks," Opt. Lett. 12, 448 (1987), pp. 448-450.

9. N.H. Farhat and B. Bai, "Phased array antenna pattern synthesis by simulated annealing," Proc. IEEE (Letters), 75 June 1987, p. 842-844.

10. N.H. Farhat, "Opto-electronic analogs of self-programming neural nets: architectures and methodologies for implementing stochastic learning by simulated annealing," App. Optics, 26, Dec. 1987, pp. 5093-5103.

11. N.H. Farhat and Z.Y. Shae, "Architectures and methodologies for self-organization and stochastic learning in opto-electronic analogs of neural nets," Proc. of the IEEE First Annual International Conference on Neural Networks, June 1987, (IEEE Cat. No. 87TH0191-7).

12. N.H. Farhat and Z.Y. Shae, "Bimodal stochastic optical learning machine," (submitted)(see preprint in Appendix VI).

13. N.H. Farhat and Z.Y. Shae, "Methods for accelerating the frame-rate of magneto-optic spatial light modulators," (submitted).

Degrees Awarded:

1. S. Miyahara, "Automated radar target recognition based on models of neural nets," Univ. of Pennsylvania, Ph.D. Dissertation, 1987.

2. K.S. Lee, "A new approach to optical information processing based on neural networks models with application to object recognition," Univ. of Pennsylvania, Ph.D. Dissertation, 1987.

Patent Disclosure:

Super-resolution - Patent disclosure filed April 9, 1987 on behalf of the University of Pennsylvania by University Patent Inc., 1465 Post Road East, P.O. Box 901, Westport, CT. 06881.

Invited Talks and Presentations:

In addition to those listed in earlier annual reports (see refs [1] and [2]), the following invited talks were presented by N. Farhat in the period Nov. 1, 1987 to May 30, 1988.

1. "Collective nonlinear optical processing based on models of neural networks," Battelle Memorial Institute, Columbus Division, Nov. 27, 1987.

2. Super-resolved recognition of radar signal targets based on models of neural networks," IEEE Philadelphia Section, February 16, 1988.

3. Self-organization and learning in neural net analogs," Philadelphia IEEE Circuits and Systems Chapter, March 2, 1988.

4. Stochastic neural nets," Long Island IEEE Computer Society, March 24, 1988.

5. "Stochastic optical learning machines," Columbia University, May 4, 1988.

Workshop Participation:

N. Farhat participated in the following workshops:

NATO Advanced Study Institute, "Electromagnetic Modeling and Measurements for Analysis and Synthesis Problems," Il Ciocco, Tuscany, Italy, Aug. 10-21, 1987.

JPL Workshop on Neural Network Devices and Applications, Jet Propulsion Laboratory, Pasadena, Feb. 18-19, 1987.

DARPA/Lincoln Lab Review Panels.
Optical Neural Nets, Caltech, Pasadena, Dec. 14, 1987
Applications, Bedford, MA., Jan. 22, 1988

NSF/ONR "Workshop on Hardware Implementation of Neuron Nets and Synapses," San Diego, CA., Jan. 13-15, 1988.

"ARO Workshop on Submillimeter Wave Imaging," Breckenridge, Colo., February 2-4, 1988.

"Neural Networks for Computing Conference," Snowbird, Utah, April 1988.

6. APPENDICES

I. Optical Implementation of Associative Memory Based on Models of Neural Networks

II. Architectures for Optoelectronic Analogs of Self-Organizing Neural Networks

III. Optoelectronic Analogs of Self-Programming Neural Nets: Architecture and Methodologies for Implementing Fast Stochastic Learning by Simulated Annealing

IV. Phased-Array Antenna Pattern Synthesis by Simulated Annealing

V. Architectures and Methodologies for Self-Organization and Stochastic Learning in Opto-Electronic Analogs of Neural Nets

VI. Biomodal Stochastic Optical Learning Machine

2.3

Optical Implementation of Associative Memory Based on Models of Neural Networks

NABIL H. FARHAT
THE MOORE SCHOOL OF ELECTRICAL ENGINEERING
UNIVERSITY OF PENNSYLVANIA
PHILADELPHIA, PENNSYLVANIA 19104

DEMETRI PSALTIS
DEPARTMENT OF ELECTRICAL ENGINEERING
CALIFORNIA INSTITUTE OF TECHNOLOGY
PASADENA, CALIFORNIA, 91125

I. Introduction

Steady progress in neuroscience and behavioral science during the last few decades is providing us with a level of understanding of the structural and functional properties of the brain that allows meaningful, albeit highly simplified, modeling of its neural networks. Several widely accepted observations can be made about the brain. It processes information in a collective manner with the aid of a network of a very large number of densely interconnected, relatively simple decision-making elements, the neurons. Each neuron makes a somewhat fuzzy decision on whether or not to change its state (firing or not firing) in the course of a computation depending on its weighted specific inputs from other neurons. It is estimated the brain has something of the order of 10^{10} to 10^{11} neurons, each making 10^3 to 10^4 connections to neighboring neurons. Information is stored by a process of electrochemical "learning" induced by representations of sensory data that

modify the weights of the synapses (Hebb, 1949). The total number of synaptic interconnections where information is stored is extremely large, approaching 10^{14} or 10^{15}.

The number of connections in a neural network is in sharp contrast to the very sparse connectivity that exists in electronic circuits of computers. This structural difference is accompanied by a distinction among the types of problems that neural nets and conventional computers are particularly good at. For instance, we know that in arithmetic operations the brain is not nearly as efficient as a digital computer; but when it comes to performing tasks of recognition, classification, and association, it can outperform even the most powerful digital computer. This difference in capabilities stems directly from the collective nature of information processing in the brain as opposed to the predominately segmented and largely serial processing architectures of conventional digital computers. The difference between the way a neural net and a digital computer approach problems is evident in simple arithmetic problems. When we are asked what 3 times 8 equals, we do not need to calculate the result, we immediately give the answer 24 because in early age we memorized multiplication tables; we simply recognize the prompt and recall the associated answer.

One of the most impressive attributes of neural information processing is its fault tolerance, which ensures graceful degradation in performance. By the age of 50 many of us may have lost a nonnegligible number of brain cells (neurons), and yet no noticeable degradation in performance and capabilities is detected. We simply learn to do better with less. There is increasing evidence that neural interconnections are a dynamic nonceasing process which may also be a factor in checking deterioration. The robustness of information processing in the brain is not confined to tolerance to failure of components but also includes an ability to tolerate a degree of error and noise in the data presented and yet continue to perform the tasks of recognition, association, or classification that may be required. In other words the brain seems to be able to reconstruct signals from partial and noisy information, which gives it a superresolving capability by which a signal or entity can be reconstructed in its entirety from a partial version. As evidence of this, note the ease with which we recognize a face in a photograph when the photograph is partially mutilated or part of it is missing. Association and superresolution are important basic operations required in pattern recognition and in the solution of ill-posed problems encountered, for example, in vision and inverse scattering. We know, for example, that signal recovery from partial information is an ill-posed problem (Tikhonov and Arsenin, 1977). This problem appears to be handled readily by neural nets because of the nonlinear nature of their decision making, their massive interconnectivity, and the feedback employed (Poggio and Koch, 1985; Farhat and Miyahara, 1986). The problem of designing memories that can be used to supplement incomplete descriptions of members of an ensemble of patterns stored in a prescribed distributed fashion has been considered by many investigators (Gabor, 1969; Willshaw *et al.*, 1969; Longuet-Higgins *et al.*, 1970; Willsaw, 1971, 1972; Nakano, 1972; Anderson *et al.*, 1977; Little, 1977; Kohonen, 1972, 1978, 1984; Amari, 1977; Grossberg, 1982; Hopfield, 1982).

The parallelism and massive interconnectivity of neural nets happen to be also the main strengths of optical processing. Hence the interest in optical implementations of neural nets (Psaltis and Farhat, 1985; Farhat *et al.*, 1985). These studies and this review are aimed at drawing attention to the fit between optics and neural modeling with the hope that neural modeling can be combined with the capabilities of modern optics, including emergent optical technologies such as high-speed nonvolatile spatial light modulators (SLMs) (Tanguay and Warde, 1980; Ross *et al.*, 1983) and optically bistable devices (OBDs) (Miller *et al.*, 1981; Gibbs *et al.*, 1981). This may lead to optical architectures and processing systems that are attractive for associative memory, pattern recognition, and machine vision systems (Guest and Gaylord, 1980; Fisher *et al.*, 1985; Athale *et al.*, 1986; Yariv and Kwong, 1986; Soffer *et al.*, 1986; Cohen, 1986). In general, optics has distinct advantages and disadvantages vis-a-vis silicon-based processing systems (Lambe *et al.*, 1985; Graph, 1985). It is widely agreed that optics provides global communication ability and dense storage for memory. Semiconductors, on the other hand, are good for building gates with which combinatorial logic circuits, the mainstay of present day computers, are constructed. Optical computers whose structure and function are motivated by neural modeling can be advantageous compared with electronic implementations of the same architectures because the connectivity, which is the most distinctive characteristic of neural nets, also happens to be the advantage of optics.

In Section II of this chapter we discuss a linear systems approach to associative memory. In Sections III and IV, optoelectronic implementations of one and two dimensional neural nets are presented, and in Section V we discuss holographic associative memories. Finally, in Section V we make general observations on the features of the network approach to optical information processing and assess the future prospects of this field.

II. Linear Systems as Associative Memories

Optical information processing systems can perform very efficiently analog linear transformations on large input data sets, and as a consequence they have been applied to a variety of signal and image processing problems

that require linear transformations. Associative memories is another application in which the ability of optical systems to perform linear transformations can prove useful. An associative memory can be implemented with a linear system by selecting the impulse response so that a desired output is produced for a set of specified inputs. If T is a matrix representing the impulse response of a system, then we require that

$$c^{(m)}\mathbf{v}^{(m)} = \mathbf{T}\mathbf{u}^{(m)} \tag{1}$$

for a set of vectors $\mathbf{u}^{(m)}$, $\mathbf{v}^{(m)}$ and constants $c^{(m)}$, the matrix T can be thought of as a memory that is addressed by the vectors $\mathbf{u}^{(m)}$ and produces the associated stored vectors $\mathbf{v}^{(m)}$. Here we are concerned with a specific class of autoassociative memory, which results when $\mathbf{u}^{(m)} = \mathbf{v}^{(m)}$ in Eq. (1). Autoassociative memories reproduce an input that exactly matches one of the stored vectors when the input is a partial or otherwise distorted version of one of the stored vectors that is sufficiently close to it. In this case Eq. (1) reduces to an eigenvalue equation, and the vectors are stored as eigenfunctions of the linear transformation T. One way this can be accomplished is by constructing the matrix T as follows:

$$\mathbf{T} = \mathbf{V}\mathbf{D}\mathbf{V}^{-1} \tag{2}$$

where

$$\mathbf{D} = \begin{pmatrix} c^{(1)} & & & & \\ & \cdots & & 0 & \\ & & c^{(M)} & & \\ & & & 0 & \\ & 0 & & \cdots & \\ & & & & 0 \end{pmatrix}$$

$$\mathbf{V} = (\mathbf{v}^{(1)} \quad \cdots \quad \mathbf{v}^{(M)} \quad \mathbf{w}^{(M+1)} \quad \cdots \quad \mathbf{w}^{(N)})$$

N is the length of the vectors v, and $M < N$ is the number of vectors stored in the $N \times N$ matrix T. The vectors w are basis vectors in the subspace that is orthogonal to the space spanned by the vectors u. Notice that in the diagonal matrix of eigenvalues D the eigenvalues are all zero for $m > M$. Thus the set of vectors w spans the null space of the linear transformation T. If the matrix T as defined in Eq. (2) is multiplied by one of the stored vectors, the same vector is reproduced at the output scaled by the constant c. An arbitrary input vector u can be written as a linear combination of the eigenfunctions of the matrix T:

$$\mathbf{u} = \sum_{m=1}^{M} d^{(m)}\mathbf{v}^{(m)} + \sum_{m=M+1}^{N} d^{(m)}\mathbf{w}^{(m)} \tag{3}$$

Let us consider the case in which all the eigenvalues are selected to be equal (i.e., $c^{(m)} = c$ for all m). Then the product of the vector u with the matrix T as defined in Eq. (3) can be written as follows:

$$\mathbf{T}\mathbf{u} = c \sum_{m=1}^{M} d^{(m)}\mathbf{v}^{(m)} \tag{4}$$

Thus the output is a linear combination of the stored vectors only. In general, if the correlation between u and a stored vector $\mathbf{v}^{(m)}$ is large, the corresponding coefficient $d^{(m)}$ is also large. Therefore the stronger contribution to the output comes from the stored datum with which it correlates best.

The role of the null space spanned by the vectors w is crucial in this scheme. If the null space were removed by making M equal to N, then any vector u would be simply replicated at the output scaled by the eigenvalue c. Clearly such a system does not perform useful computation. In general the fidelity with which the stored datum that best matches u is produced decreases monotonically with the ratio (M/N). This imposes the limit on the storage capacity of the memory. We discuss the issue of storage capacity in more detail later.

If the matrix T is formed by summing the outer products of the vectors v with u (or the outer products of each vector v with itself in the case of autoassociations), then Eq. (1) is satisfied approximately as long as M is sufficiently smaller than N. The outer product scheme has been used extensively because of its simplicity and bypassing of the need to invert a matrix even though it is not in principle as powerful as the true eigensystem approach. Moreover, the outer product scheme has been proposed as a model for the way information may be stored in neural networks; basically it is a linear approximation to Hebb's hypothesis about learning (Hebb, 1949), and thus much of the work on associative memories that has come from neural network modeling deals with the outer product scheme. For the case of autoassociative memory the matrix T formed as the sum of outer products is given by

$$\mathbf{T} = \sum_{m=1}^{M} \mathbf{v}^{(m)}\mathbf{v}^{(m)t} \tag{5}$$

where $\mathbf{v}^t$ is the transpose of v. The product of a vector u with T is also a linear combination of the stored vectors:

$$\mathbf{T}\mathbf{u} = \sum_{m=1}^{M} d^{(m)}\mathbf{v}^{(m)} \tag{6}$$

with $d^{(m)} = \mathbf{v}^{(m)t}\mathbf{u}$. In this case there is no guarantee that the output will be one of the stored vectors even if the input u is one of the stored vectors.

However, as before, the output is a linear combination of the stored vectors, and the vector that correlates best with the input is amplified the most.

In general it is desirable that an associative memory produce at its output the stored vector most closely associated with the input **u** (i.e., perform a closest-neighbor search). The linear models discussed produce a linear combination of all the memories with the correct stored vector emphasized. Therefore they do not produce the correct result but an approximation to it; the desired vector can be thought of as the signal term and the rest of the vectors as an additive interference that produces a noisy estimate of the stored datum. If the input and stored vectors are restricted to binary values, then it is possible to restore the output of the linear memory to the stored binary vector simply by thresholding. Consider Eq. (6) which represents the output of the linear system in the outer product scheme. We require that

$$\mathrm{sgn}[\mathbf{Tu}] = \mathrm{sgn}\left[\sum_{m=1}^{M} d^{(m)}\mathbf{v}^{(m)}\right] \tag{7}$$

where $\mathrm{sgn}[x] = 1$ if $x > 0$ and -1 otherwise. If Eq. (7) is satisfied, then the addition of thresholding to the linear associative memory yields the precise binary stored vector that has the maximum correlation with the input vector **u**. If we knew the precise conditions under which the condition in Eq. (7) is satisfied then we would have a complete characterization of the performance of this type of memory. We do not yet have the complete answer to this question. We know that if the input vector **u** is one of the stored vectors $\mathbf{v}^{(m)}$, then condition (7) is satisfied if $M < N/4 \ln N$ (Venkatesh and Psaltis, 1985; McEliece *et al.*, 1986). This gives an estimate of the storage capacity for the outer product scheme, defined as the maximum number of binary vectors that can be stored in the matrix **T** such that the output is equal to the input vector for all the stored vectors.

When **u** is not exactly equal to one of the stored binary vectors, then the output of the system is in general a binary-valued vector that is an approximation to the stored vector that correlates best with the input. We can use this estimate as the input to the memory and obtain a new estimate that is in general a better approximation to the stored vector. When this procedure is repeated a sufficient number of times, the correct stored vector is produced as an output. If the storage capacity is not exceeded, then once one of the correct outputs is produced it keeps reproducing itself (i.e., a stable point is reached). Hopfield (1982) has shown that, when the vector **T** is symmetric, the outer product formulation that guarantees stability is reached; Venkatesh and Psaltis (1985) have shown that this is also true in almost all cases for the eigenfunction scheme. The radius of attraction around a stored vector is defined as the maximum number of bits that an input vector can differ on the average from a stored vector and still produce, as a stable output,

the correct association. The number of locations at which two binary vectors differ is the Hamming distance between them. Clearly this measure is equally important to the storage capacity. To our knowledge a strong theoretical result for the radius of attraction does not exist. In the eigenfunction scheme it is relatively easy to see, however, that there is a direct inverse relationship between the storage capacity and the radius of attraction. M in this case can be as high as N if the stored vectors are linearly independent. However, at this extreme all input vectors simply reproduce themselves at the output. Thus the radius of attraction is zero for all stored vectors. If $M = 1$, then any input vector converges to the stored word or its complement; thus the radius of attraction is the maximum N bits in this case. Of course the cases of interest lie in between. A single metric that can be used in evaluating the performance of these memories is the product of the radius of attraction plus one and the storage capacity. In computer simulations it has been found that this product can exceed N (its value at the two extreme cases discussed above), but we do not know where the optimum operating point is.

III. Optoelectronic Implementation of One-Dimensional Neural Nets

The outer-product method for storing binary vectors in an associative memory, discussed in the previous section, is now considered in more detail in terms of the components of the stored vectors. Given a set of M bipolar, binary $(1, -1)$ vectors $v_i^{(m)}$ for $i = 1, 2, 3, \ldots, N$, $m = 1, 2, 3, \ldots, M$, these are stored in a matrix **T** in accordance to the outer-product recipe,

$$T_{ij} = \sum_{m=1}^{M} v_i^{(m)} v_j^{(m)} \qquad i, j = 1, 2, 3, \ldots, N, \qquad T_{ii} = 0 \tag{8}$$

If the memory is addressed by multiplying the matrix **T** with one of these vectors, say $\mathbf{v}^{(m0)}$, it yields the estimate

$$\hat{v}_i^{(m0)} = \sum_{j=1}^{N} T_{ij} v_j^{(m0)} = \sum_{j}^{N}\sum_{m}^{M} v_i^{(m)} v_j^{(m)} v_j^{(m0)} \tag{9}$$

$$= (N-1) v_i^{(m0)} + \sum_{m \neq m0}^{M} a^{(m)} v_i^{(m)} \tag{10}$$

where

$$a^{(m)} = \sum_{j=1}^{N} v_j^{(m)} v_j^{(m0)}$$

and $\hat{\mathbf{v}}^{(m0)}$ consists of the sum of two terms: The first is the input vector amplified by $(N-1)$; the second is a linear combination of the remaining

stored vectors and represents an unwanted cross talk term. The value of the coefficients $a^{(m)}$ is equal to $(N-1)^{1/2}$ on the average (the standard deviation of the sum of $N-1$ random bits), and since $(M-1)$ such coefficients are randomly added, the value of the second term is, on the average, equal to $[(M-1)(M-1)]^{1/2}$. If N is sufficiently larger than M, then with high probability the elements of the vector $\hat{\mathbf{v}}^{(m0)}$ will be positive if the corresponding elements of $\mathbf{v}^{(m0)}$ are equal to $+1$, and negative otherwise. Thresholding of $\hat{\mathbf{v}}^{(m0)}$ will therefore return $\mathbf{v}^{(m0)}$:

$$v_i^{(m0)} = \operatorname{sgn}\left(\sum_j^N T_{ij} v_j^{(m0)}\right) \tag{11}$$

When the memory is addressed with a binary-valued vector that is not one of the stored words, then the vector-matrix multiplication and thresholding operation yield an output binary-valued vector that, in general, is an approximation of the stored word that is at the shortest Hamming distance from the input vector. As pointed out earlier, if this output vector is fed back and used as the input to the memory, the new output is generally a more accurate version of the stored word; and continued iteration converges to the correct vector.

The insertion and readout of memories using the outer-product scheme is depicted schematically in Fig. 1. Note that in Fig. 1b the estimate $\hat{\mathbf{v}}^{(m0)}$ can be viewed as the weighted projection of T. Recognition of an input vector that corresponds to one of the state vectors of the memory or is close to it (in the Hamming sense) is manifested by a stable state of the system. In practice unipolar binary (0, 1) vectors or words $\mathbf{b}$ of bit length N may be of interest. The above equations are then applicable with $(2b_i - 1)$ replacing v_i in Eq. (8) and b_i replacing v_i in Eq. (9). For such vectors the

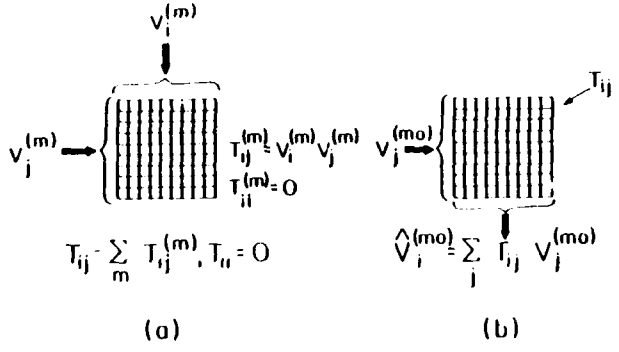


Fig. 1. Insertion (a) and readout (b) of memories

SNR of the estimate $\hat{\mathbf{v}}^{(m0)}$ can be shown to be lower by a factor of the square root of two (Hopfield, 1982).

An example of the T matrix formed from four binary unipolar vectors, each being $N = 20$ bits long, is given in Fig. 2 along with the result of a numerical simulation of the process of initializing the memory matrix with a partial version of $b_i^{(4)}$ in which the first eight digits of $b_i^{(4)}$ are retained and the rest are set to zero. The Hamming distance between the initializing vector and $b_i^{(4)}$ is six bits, and it is nine or more bits for the other three stored vectors. It is seen that the partial input is recognized as $b_i^{(4)}$ in the third iteration, and the output remains stable as $b_i^{(4)}$ thereafter. This convergence to a stable state generally persists even when the T matrix is binarized or clipped by replacing negative elements by minus ones and positive elements by plus ones evidencing the robustness of the content addressable memory (CAM's). A binary synaptic matrix* has the practical advantage of being more readily implementable with fast programmable spatial light modulators (SLM) with storage capability such as the Litton LIGHTMOD (Ross *et al.*, 1983). Such a binary matrix, implemented photographically, is used in the optical implementation described below.

Several schemes for optical implementation of a CAM based on the outlined principle have been described. In one of the implementations (Psaltis and Farhat, 1985) an array of light-emitting diodes (LEDs) is used to represent the logic elements or neurons of the network. Their state (on or off) can represent unipolar binary vectors such as the state vectors $b_i^{(m)}$ that are stored in the memory matrix T_{ij}. Global interconnection of the elements is realized as shown in Fig. 3a through the addition of nonlinear feedback (thresholding, gain, and feedback) to a conventional optical vector matrix multiplier (Goodman *et al.*, 1978) in which the array of LEDs represents the input vector, and an array of photodiodes (PD) is used to detect the output vector. The output is thresholded and fed back in parallel to drive the corresponding elements of the LED array. Multiplication of the input vector by the T_{ij} matrix is achieved by horizontal imaging and vertical smearing of the input vector that is displayed by the LEDs on the plane of the T_{ij} mask (by means of an anamorphic lens system omitted from Fig. 3a for simplicity). A second anamorphic lens system (also not shown) is used to collect the light emerging from each row of the T_{ij} mask on individual photosites of the PD array. A bipolar T_{ij} matrix is realized in incoherent light by dividing each row of the T_{ij} matrix into two subrows, one for positive and one for negative values, and bringing the light emerging from each subrow to focus on two adjacent photosites of the PD array that are electrically connected in opposition as depicted in Fig. 3b. In the system

*Strictly speaking such a matrix is ternary as it also contains zeros.

$b^{(1)}$	$b^{(2)}$	$b^{(3)}$	$b^{(4)}$
1	0	1	0
0	1	1	1
1	0	0	1
1	1	0	1
0	1	0	0
1	0	1	1
1	0	0	1
1	1	1	0
0	0	1	1
1	0	1	0
1	0	0	1
1	0	1	0
0	1	1	0
1	1	0	0
0	1	1	1
0	0	1	0
1	0	1	1
0	0	1	1
0	1	0	1
1	1	0	0

(a)

0	−2	0	−2	−2	2	0	2	0	4	0	4	0	0	−2	2	2	0	−4	0
−2	0	−2	0	0	0	−2	0	2	−2	−2	−2	2	−2	4	0	0	2	2	−2
0	−2	0	2	−2	2	4	−2	0	0	4	0	−4	0	−2	−2	2	0	0	0
−2	0	2	0	0	0	2	0	−2	−2	2	−2	−2	2	0	−4	0	−2	2	2
−2	0	−2	0	0	−4	−2	0	−2	−2	−2	−2	2	2	0	0	4	−2	2	2
2	0	2	0	−4	0	2	0	2	2	2	2	−2	−2	0	0	4	2	−2	−2
0	−2	4	2	−2	2	0	2	0	0	4	0	−4	0	−2	−2	2	0	0	0
2	0	−2	0	0	0	−2	0	−2	2	−2	2	2	2	0	0	0	−2	−2	2
0	2	0	−2	−2	2	0	−2	0	0	0	0	0	−4	2	2	2	4	0	−4
4	−2	0	−2	−2	2	0	2	0	0	0	4	0	0	−2	2	2	0	−4	0
0	−2	4	2	−2	2	4	−2	0	0	0	0	−4	0	−2	−2	2	0	0	0
4	−2	0	−2	−2	2	0	2	0	4	0	0	0	0	−2	2	2	0	−4	0
0	2	−4	−2	2	−2	−4	2	0	0	−4	0	0	0	2	2	−2	0	0	0
0	−2	0	2	2	−2	0	2	−4	0	0	0	0	−2	−2	−2	−4	0	4	
−2	4	−2	0	0	0	−2	0	2	−2	−2	−2	2	−2	0	0	0	2	2	−2
2	0	−2	−4	0	0	−2	0	2	2	−2	2	2	−2	0	0	0	2	−2	−2
2	0	2	0	−4	4	2	0	2	2	2	2	−2	−2	0	0	0	2	−2	−2
0	2	0	−2	−2	2	0	−2	4	0	0	0	0	−4	2	2	2	0	0	−4
−4	2	0	2	2	−2	0	−2	0	−4	0	−4	0	0	2	−2	−2	0	0	0
0	−2	0	2	2	−2	0	2	−4	0	0	0	0	4	−2	−2	−2	−4	0	0

(b)

0	1	1	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0		Partial input ($b^{(4)}$)
−2	−4	6	4	−6	4	6	−4	2	−2	10	−2	−10	−2	0	−8	8	2	2	−2	1st Estimate
0	0	1	1	0	1	1	0	1	0	1	0	0	0	0	0	1	1	1	0	1st Thresholding
0	0	1	1	0	1	1	0	1	0	1	0	0	0	1	0	1	1	0	0	2nd Iteration
0	1	1	1	0	1	1	0	1	0	1	0	0	0	1	0	1	1	1	0	3rd Iteration
0	1	1	1	0	1	1	0	1	0	1	0	0	0	1	0	1	1	1	0	4th Iteration

(c)

Fig. 2. Numerical example of supplementing a partial input. $N = 20$, $m = 4$. (a) Stored vectors. (b) Interconnection or synaptic matrix. (c) Results of initializing with a partial version of (b) showing convergence to a stable state on the third iteration.

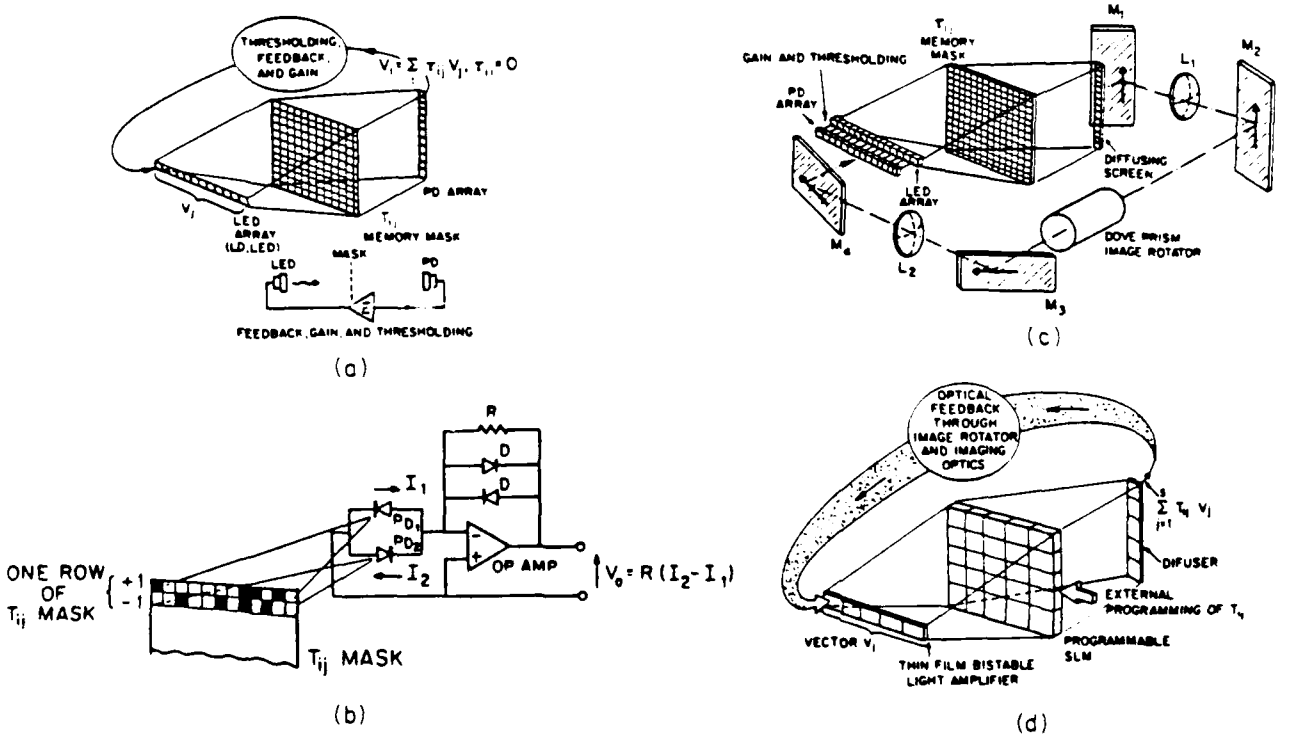


Fig. 3. Architectures for optical implementation of a content-addressable memory based on models of neural nets. (a) Matrix vector multiplier incorporating nonlinear electronic feedback. (b) optoelectronic scheme for realizing binary bipolar mask transmittance in incoherent light. (c) optical feedback scheme incorporating hybrid optical light amplifier array. (d) optical feedback with thin film bistable light amplifier and programmable connectivity matrix.

shown in Fig. 3a, feedback is achieved by electronic wiring. It is possible and preferable to dispose of electronic wiring altogether and replace it by optical feedback. This can be achieved by combining the PD and LED arrays in a single compact hybrid or monolithic structure that can also be made to contain all IC's for thresholding, amplification, and driving of LEDs. Optical feedback is particularly interesting when we consider that arrays of nonlinear optical light amplifiers with internal feedback (Porada, 1983), arrays of nonlinear optical switches (Gibbs *et al.*, 1981; Miller *et al.*, 1984) can be used to replace the PD/LED array. This can lead to simple CAM structures that may be interconnected to perform higher-order computations than the nearest-neighbor search performed by a single CAM.

A variation of the scheme presented in Fig. 3a was constructed to simulate optically a network of $N = 32$ neurons. The system, details of which are given below, was constructed with an array of 32 LEDs and two multichannel silicon PD arrays, each consisting of 32 elements. Twice as many PD elements as LEDs are needed to implement a bipolar memory mask transmittance in incoherent light in accordance to the scheme of Fig. 3b. A bipolar binary T_{ij} mask was prepared for $M = 3$ binary state vectors. The three vectors or words chosen, their Hamming distances from each other, and the resulting T_{ij} memory matrix are shown in Fig. 4. The mean Hamming distance between the three vectors is 16. A binary photographic transparency of 32×64 square pixels was computer-generated from the T_{ij} matrix by assigning the positive values in any given row of T_{ij} to transparent pixels in one subrow of the mask and the negative values to transparent pixels in the adjacent subrow. To insure that the image of the input LED array is uniformly smeared over the memory mask, it was found convenient to split the mask into two halves, as shown in Fig. 5, and to use the resulting submasks in two identical optical arms, as shown in Fig. 6. The size of the subrows of the memory submasks was made exactly equal to the element size of the PD arrays in the vertical direction, which were placed in register against the masks. Light emerging from each subrow of a memory submask was collected (spatially integrated) by one of the vertically oriented elements of the multichannel PD array. In this fashion the anamorphic optics required in the output part of Fig. 3a are disposed of, resulting in a simpler and more compact system. Further simplification can be achieved similarly by using an array of light-emitting line sources instead of the LEDs, which would dispose of the input optics and result in the compact associative memory chip concept depicted in Fig. 7. Considerable versatility can be added to such a chip by employing a computer-controlled dynamic nonvolatile SLM as a programmable synaptic mask as shown pictorially in Fig. 3d. Pictorial views of the input LED array and the two submask-PD array assemblies are shown in Figs. 8a and 8b, respectively. In Fig. 8b the left

Stored words:

```
Word 1 : 1 1 1 0 0 0 0 1 0 1 0 1 1 1 0 1 1 0 1 1 1 1 0 1 1 0 0 0 0 0 1 0
Word 2 : 0 1 1 0 0 0 0 0 0 0 1 0 0 1 0 1 0 1 0 0 1 1 1 1 0 1 0 1 1 0 1 0
Word 3 : 1 0 1 1 0 0 1 1 1 1 1 1 1 1 1 0 0 0 1 0 1 1 0 0 0 0 1 1 0 0 0 0
```

Hamming distance from word to word:

WORD	1	2	3
1	0	15	14
2	15	0	19
3	14	19	0

Clipped memory matrix:

```
 0 -1  1  1 -1 -1  1  1  1  1 -1  1  1  1  1 -1  1 -1  1  1  1  1 -1 -1  1 -1  1 -1 -1 -1 -1
-1  0  1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1  1 -1  1  1  1 -1  1  1  1  1  1  1 -1 -1  1 -1  1 -1
 1  1  0 -1 -1 -1  1 -1  1  1  1  1  1 -1  1 -1 -1  1 -1  1  1 -1  1 -1 -1 -1  1 -1 -1  1 -1
 1 -1 -1  0  1  1  1  1  1  1  1  1 -1  1 -1 -1 -1 -1  1 -1 -1 -1 -1 -1 -1  1  1 -1  1 -1  1
-1 -1 -1  1  0  1 -1  1 -1 -1 -1 -1 -1  1 -1  1  1 -1  1 -1 -1  1 -1  1  1 -1  1  1 -1  1  1
-1 -1 -1  1  1  0  1 -1  1 -1 -1 -1 -1 -1  1 -1  1  1 -1  1 -1 -1  1 -1  1  1 -1  1  1 -1  1
 1 -1 -1  1  1  1  0  1  1  1 -1  1  1  1 -1  1 -1 -1 -1  1 -1 -1 -1 -1 -1 -1  1  1 -1  1 -1
 1 -1  1  1 -1 -1  1  0  1  1 -1  1  1  1  1 -1  1 -1  1  1  1  1 -1 -1  1 -1  1 -1 -1 -1 -1
 1 -1 -1  1  1  1  1  1  0 -1  1  1  1  1 -1  1 -1 -1 -1 -1 -1 -1 -1 -1 -1  1  1 -1  1 -1  1
 1 -1  1  1 -1 -1  1  1  1  0 -1  1  1  1 -1  1 -1  1  1  1  1 -1 -1  1 -1  1 -1 -1 -1 -1 -1
-1 -1  1  1 -1 -1  1 -1  1 -1  0 -1 -1  1  1 -1 -1  1 -1 -1  1  1  1 -1 -1  1  1  1  1 -1 -1
 1 -1  1  1 -1 -1  1  1  1  1 -1  0  1  1  1 -1  1 -1  1  1  1  1 -1 -1  1 -1  1 -1 -1 -1 -1
 1 -1  1  1 -1 -1  1  1  1  1 -1  1  0  1  1 -1  1 -1  1  1  1  1 -1 -1  1 -1  1 -1 -1 -1 -1
 1  1  1 -1 -1 -1 -1  1 -1  1  1  1  1  0 -1  1 -1 -1  1 -1  1  1 -1  1 -1 -1 -1  1 -1 -1  1 -1
 1 -1 -1  1  1  1  1  1  1  1  1  1  1 -1  0 -1 -1 -1  1 -1 -1 -1 -1 -1 -1 -1  1  1 -1  1 -1  1
-1  1  1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1  1 -1  0  1  1 -1  1  1  1  1  1  1 -1 -1  1 -1  1 -1
 1  1 -1 -1  1  1 -1  1 -1  1 -1  1  1 -1 -1  1  0 -1  1  1 -1 -1 -1  1  1 -1 -1 -1 -1  1  1  1
-1  1 -1 -1  1  1  1 -1 -1 -1 -1 -1 -1  1 -1  1 -1  0 -1 -1 -1 -1  1  1 -1  1  1  1  1  1  1  1
 1 -1  1  1 -1 -1  1  1  1  1 -1  1  1  1  1 -1  1 -1  0  1  1  1 -1 -1  1 -1  1 -1 -1 -1 -1 -1
 1  1 -1 -1  1  1 -1  1 -1  1 -1  1  1 -1 -1  1 -1  1 -1  0 -1 -1 -1  1  1 -1 -1 -1 -1  1  1  1
 1  1  1 -1 -1 -1 -1  1 -1  1  1  1  1  1 -1  1 -1  1 -1  1  0 -1  1 -1 -1 -1 -1  1 -1 -1  1 -1
 1  1  1 -1 -1 -1 -1  1 -1  1  1  1  1  1 -1  1 -1  1 -1  1  1  0 -1  1 -1 -1 -1  1 -1 -1  1 -1
-1  1 -1 -1  1  1 -1 -1 -1 -1  1 -1 -1 -1 -1  1 -1  1 -1 -1 -1 -1  0  1 -1  1 -1  1  1  1  1  1
-1  1  1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1  1 -1  1  1  1  1 -1  1  1  1  0  1  1 -1 -1  1 -1  1 -1
 1  1 -1 -1  1  1 -1  1 -1  1 -1  1  1 -1 -1  1 -1  1 -1  1 -1 -1 -1 -1  1  0 -1 -1 -1  1  1  1
-1  1 -1 -1  1  1  1 -1 -1 -1  1 -1 -1 -1 -1  1 -1  1 -1 -1 -1 -1  1  1 -1  0 -1  1  1  1  1  1
 1 -1 -1  1  1  1  1  1  1  1  1  1  1 -1  1 -1 -1 -1  1 -1 -1 -1 -1 -1 -1 -1  0  1 -1  1 -1  1
-1 -1  1 -1 -1  1 -1  1 -1  1 -1 -1 -1  1  1 -1 -1  1 -1  1 -1 -1  1  1  1 -1 -1  1  0  1 -1 -1
-1  1 -1 -1  1  1 -1 -1 -1 -1  1 -1 -1 -1 -1  1 -1  1 -1 -1 -1 -1 -1  1  1 -1  1 -1  1  0 -1  1
-1 -1  1  1 -1 -1 -1 -1 -1 -1 -1 -1 -1  1 -1  1  1  1 -1  1  1  1  1  1  1  1 -1 -1  1 -1  0 -1
-1 -1 -1  1  1  1 -1  1 -1 -1 -1 -1 -1 -1  1 -1  1  1 -1  1 -1 -1  1 -1  1  1  1 -1  1  1 -1  0
```

Fig. 4. Three stored words, their Hamming distances, and their clipped memory matrix.

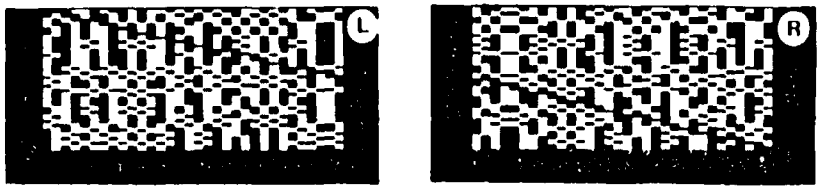
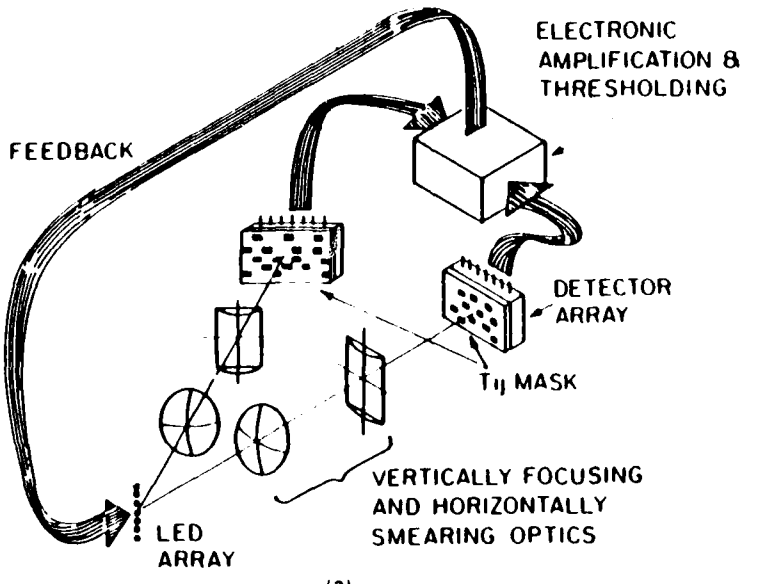


Fig. 5. Two halves of binary memory mask.

memory submask-PD array assembly is shown with the submask removed to reveal the silicon PD array situated behind it. All electronic circuits (amplifiers, thresholding comparators, LED drivers, etc.) in the 32 parallel feedback channels are contained in the electronic amplification and thresholding box shown in Fig. 6a and in the boxes on which the LED array and the two submask-PD array assemblies are mounted (see Fig. 8). A pictorial view of a composing and display box is shown in Fig. 9. This contains an arrangement of 32 switches and a 32-element LED display panel whose elements are connected in parallel to the input LED array. The function of this box is to compose and display the binary input word or vector that appears on the input LED array of the system shown in Fig. 8a. Once an input vector is selected it appears displayed on the composing box and on the input LED box simultaneously. A single switch is then thrown to operate the system with the composed vector as the initializing vector. The final state of the system, the output, appears after a few iterations, displayed on the input LED array and the display box simultaneously. This procedure provides for convenient exercising of the system to study its response versus stimulus behavior. An input vector is composed, and its Hamming distance from each of the nominal state vectors stored in the memory is noted. The vector is then used to initialize the CAM, as described above, and the output vector representing the final state of the CAM, appearing almost immediately on the display box, is noted. The response time of the electronic feedback channels as determined by the 3-dB roll-off of the amplifiers was about 60 ms. Speed of operation was not an issue in this study, and thus low response time was chosen to facilitate hardware implementation.

The results of exercising and evaluating the performance of the system are tabulated in Table I. The first run of initializing vectors used in exercising the system were error-laden versions of the first word $b_1^{(1)}$. These were obtained from $b_1^{(1)}$ by successively altering (switching) the states of 1, 2, 3, ... up to N of its digits starting from the Nth digit. In doing so, the Hamming distance between the initializing vector and $b_1^{(1)}$ is increased linearly in unit steps as shown in the first column of Table I, whereas, on the average, the



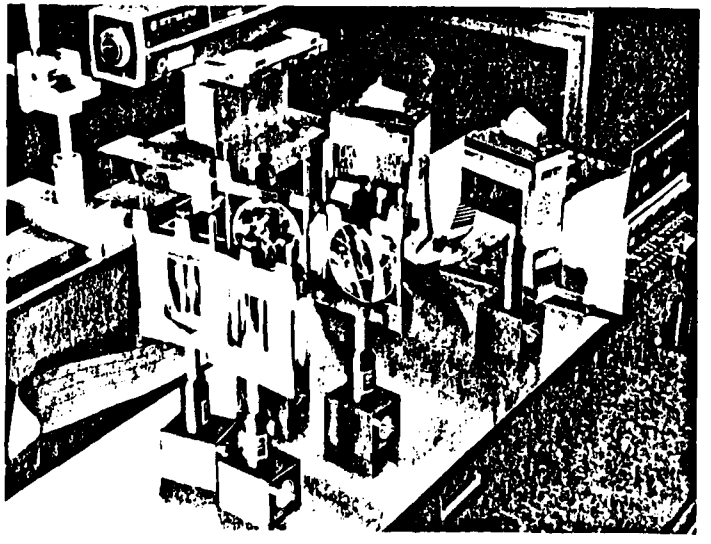


Fig. 6. Arrangement for optical implementation of content addressable memory. (a) Optoelectronic circuit diagram, (b) pictorial view.

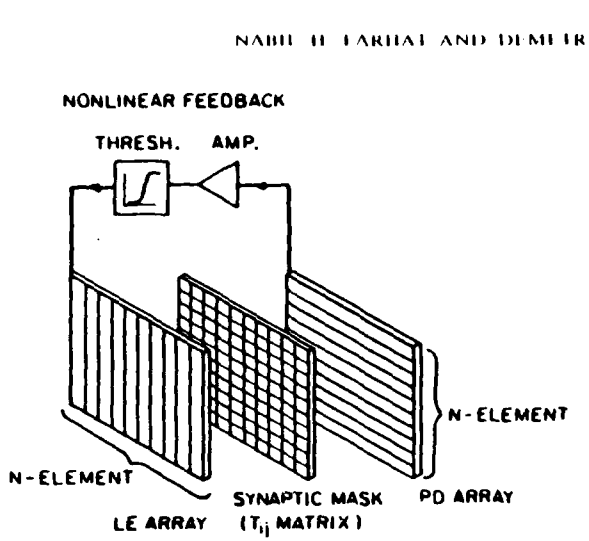


Fig. 7. Concept of optoelectronic associative-memory chip (exploded view)

Hamming distance between all these initializing vectors and the other two state vectors remained roughly the same, about $N/2 - 16$. The final states of the memory, that is, the steady-state vectors displayed at the output of the system (the composing and display box) when the memory is prompted by the initializing vectors are listed in column 2 of Table I. When the Hamming distance of the initializing vector from $b_i^{(1)}$ is less than 11, the input is always recognized correctly as $b_i^{(1)}$. The CAM is able therefore to recognize the input vector even when up to 11 of its digits (37.5%) are wrong. This performance is identical to the results obtained with a digital simulation, shown in parentheses in column 1 for comparison. When the Hamming distance is increased further to values lying between 12 and 22, the CAM is confused and identifies erroneously other state vectors, mostly $b_i^{(3)}$, as the input. In this range the Hamming distance of the initializing vectors from any of the stored vectors is roughly equal, making it more difficult for the CAM to decide. Note that the performance of the CAM and the resulting digital simulation in this range of Hamming distance are comparable except for the appearance of oscillations (designated by OSC) in the digital simulation when the outcome oscillated among several vectors that were not the nominal state vectors of the CAM. Beyond a Hamming distance of 22 both the optical system and the digital simulation identified the initializing vectors as the complement of $b_i^{(1)}$. This is expected because it can be shown using Eq. (1) that the T_{ij} matrix formed from a set of vectors is identical to that formed by the complementary set of the same

Fig. 8. (a) Pictorial views of input LED array and (b) memory submask/photodetector array assemblies

vectors. The complementary vector can be viewed as a contrast-reversed version of the original vector.

Similar results of initializing the CAM with error-laden versions of $b_i^{(2)}$ and $b_i^{(3)}$ were also obtained. These are presented in columns 2 and 3 of Table I. Here again we see that when the Hamming distance of the initializing vector from $b_i^{(3)}$ ranged between 1 and 14, the CAM recognized the input correctly as $b_i^{(3)}$ as shown in column 3 of the table; and as such it did

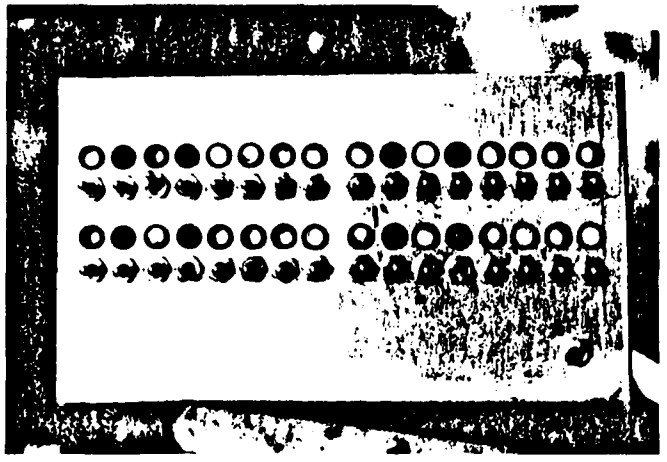


Fig. 9. Word composer and final-state display box.

slightly better than the results of digital simulation. Oscillatory behavior is also observed here in the digital simulation when the range of Hamming distances between the initializing vector and all stored vectors approached the mean Hamming distance between the stored vectors. Beyond this range the memory recognizes the input as the complementary of $b_i^{(3)}$.

In studying the results presented in Table I, *one can make several* observations: The optically implemented CAM is working as accurately as the digital simulations and perhaps more accurately if we consider the absence of oscillations, which are believed to be suppressed in the system because of the smooth thresholding performed by the nonlinear transfer function of the electronic amplifiers compared with the sharp thresholding in digital computations. The smooth nonlinear transfer function and the *finite time constant of the optical system provide a relaxation mechanism* that substitutes for the role of the asynchronous switching that is required to provide convergence to a stable state (Hopfield, 1982). Generally the system is able to conduct successful nearest-neighbor searches when the inputs to the system are versions of the nominal state vectors containing up to about 30% error in their digits. It is worth noting that this performance is achieved in a system built from off-the-shelf electronic and optical components and with relatively little effort in optimizing and fine tuning the system for improved accuracy, thereby confirming the fact that accurate overall computation can be performed with relatively inaccurate individual components. It is also worth noting that despite accidental failure of two elements of the PD array, the net continued to perform as described in

TABLE I

OPTICAL CAM PERFORMANCE[a,b]

Hamming distance of initializing vector from $b_i^{(m)}$	Recognized vector ($m-1$)	Recognized vector ($m-2$)	Recognized vector ($m-3$)
0	1 (1)	2 (2)	3 (3)
1	1 (1)	2 (2)	3 (3)
2	1 (1)	2 (2)	3 (3)
3	1 (1)	2 (2)	3 (3)
4	1 (1)	2 (2)	3 (3)
5	1 (1)	2 (2)	3 (3)
6	1 (1)	2 (2)	3 (3)
7	1 (1)	2 (2)	3 (3)
8	1 (1)	2 (2)	3 (3)
9	1 (1)	2 (2)	3 (3)
10	1 (1)	1 (1)	3 (3)
11	1 (1)	2 (2)	3 (3)
12	3 (3)	3, 2 ($\bar{3}$)	3 (3)
13	3 (3)	3 ($\bar{3}$)	3 (2)
14	3 (3)	1, 3 ($\bar{1}$)	3 ($\bar{2}$)
15	1 (OSC)	1 (1)	2, 3 ($\bar{2}$)
16	3 (OSC)	1 (1)	$\bar{2}$ ($\bar{2}$)
17	3 (OSC)	1 (OSC)	$\bar{2}$ ($\bar{2}$)
18	3 (3)	1 ($\bar{2}$)	3 (OSC)
19	3 ($\bar{2}$)	$\bar{2}$ ($\bar{2}$)	$\bar{2}$ ($\bar{2}$)
20	3 ($\bar{1}$)	$\bar{2}$ ($\bar{2}$)	$\bar{2}$ (OSC)
21	1, 2 ($\bar{1}$)	$\bar{2}$ ($\bar{2}$)	$\bar{3}$ (OSC)
22	3 ($\bar{1}$)	$\bar{2}$ ($\bar{2}$)	$\bar{3}$ (OSC)
23	1 ($\bar{1}$)	$\bar{2}$ ($\bar{3}$)	$\bar{3}$ (OSC)
24	1 (1)	2 ($\bar{2}$)	3 (3)
25	1 (1)	$\bar{2}$ ($\bar{2}$)	3 ($\bar{3}$)
26	1 (1)	$\bar{2}$ ($\bar{2}$)	$\bar{3}$ ($\bar{3}$)
27	1 (1)	$\bar{2}$ ($\bar{2}$)	$\bar{3}$ ($\bar{3}$)
28	1 (1)	$\bar{2}$ ($\bar{2}$)	$\bar{3}$ ($\bar{3}$)
29	1 (1)	$\bar{2}$ ($\bar{2}$)	$\bar{3}$ ($\bar{3}$)
30	1 (1)	$\bar{2}$ ($\bar{2}$)	3 (3)
31	1 (1)	$\bar{2}$ ($\bar{2}$)	$\bar{3}$ ($\bar{3}$)
32	1 (1)	2 ($\bar{2}$)	$\bar{3}$ ($\bar{3}$)

[a] Values in parentheses are results of digital simulation.
[b] Values with overbars designate complementary vectors.

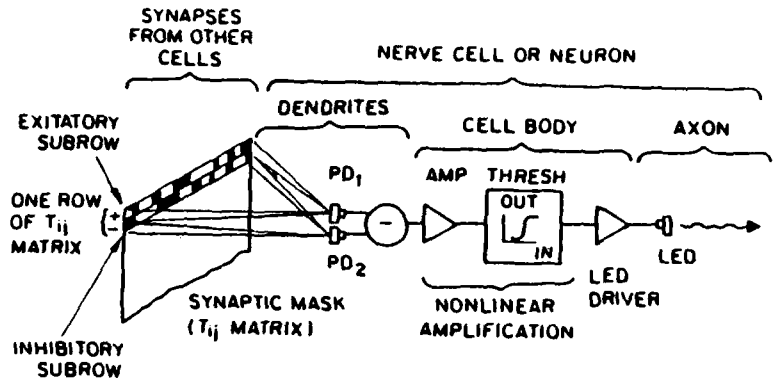


Fig. 10. Optoelectronic analog of a single nerve cell.

Table I (i.e., no noticeable change in performance was observed). In the above optoelectronic implementation, an artificial neuron can be divided into components that can be identified with their biological counterpart as shown in Fig. 10.

IV. Optoelectronic Implementation of Two-Dimensional Neural Nets

We now examine optoelectronic methods for storing and retrieving information arranged in two-dimensional format (e.g., images). Compatibility with two-dimensional data formats may be of practical interest in applications such as machine vision, and the potential exists for the optical implementation of networks containing larger numbers of neurons if they are arranged in two dimensions. The scheme that is described (Farhat and Psaltis, 1985) is a direct extension of the procedure for formation and readout of memories in the one-dimensional architecture discussed in the preceding section and summarized in Fig. 11. Given a set of two-dimensional bipolar binary patterns $v_{ij}^{(m)}$ with $i = 1, \ldots, N$, $j = 1, \ldots, N$, and $m = 1, \ldots, N$, these can be stored in an outer-product associative memory in the following manner: For each element of a pattern $v^{(m)}$ a new $N \times N$ matrix is formed by multiplying the value of this element with all the remaining elements of the matrix and setting the self-product equal to zero. The outcome is a new set of $N \times N$ binary bipolar matrices each having $N \times N$ elements. A formal description of this procedure is

$$T_{ijkl}^{(m)} = v_{ij}^{(m)} v_{kl}^{(m)} \tag{12}$$

which is a four-dimensional matrix. The overall memory matrix is formed

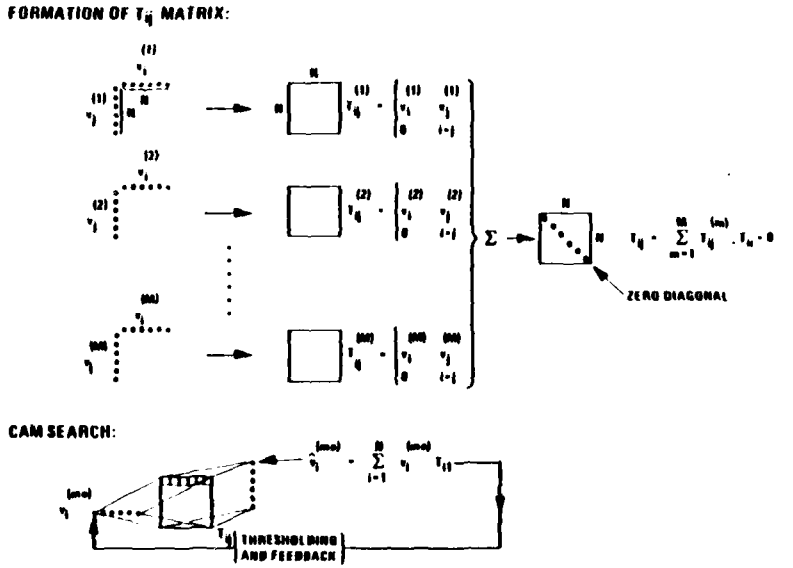


Fig. 11. Pictorial representation of formation and search of a two-dimensional connectivity matrix for a one-dimensional neural net

by adding all matrices $T_{ijkl}^{(m)}$:

$$T_{ijkl} = \sum_{m=1}^{M} v_{ij}^{(m)} v_{kl}^{(m)} \tag{13}$$

A schematic representation of these procedures is given in Fig. 12.

Two-dimensional unipolar binary entities $b_{ij}^{(m)}$ are of practical interest. As in the one-dimensional case, these can be transformed into bipolar binary matrices by setting $v_{ij}^{(m)} = (2b_{ij}^{(m)} - 1)$ which are then used to form the T_{ijkl}. Also as in the one-dimensional case, the prompting entity can be unipolar binary $b_{ij}^{(m)}$, which would simplify optical implementations that use incoherent light.

Architectures for optical implementation of two-dimensional networks must contend with the task of realizing a fourth-rank memory matrix. Here a scheme is presented that is based on the partitioning of the four-dimensional matrix into an array of two-dimensional $N \times N$ submatrices.

Information is retrieved from this memory by forming, in a manner similar to Eq. (9), the product of an input $b_{kl}^{(m0)}$ with the four-dimensional matrix:

$$\hat{b}_{ij}^{(m0)} = \sum_{k,l}^{N} T_{ijkl} b_{kl}^{(m0)} \qquad i, j, k, l = 1, 2, \ldots, N \tag{14}$$

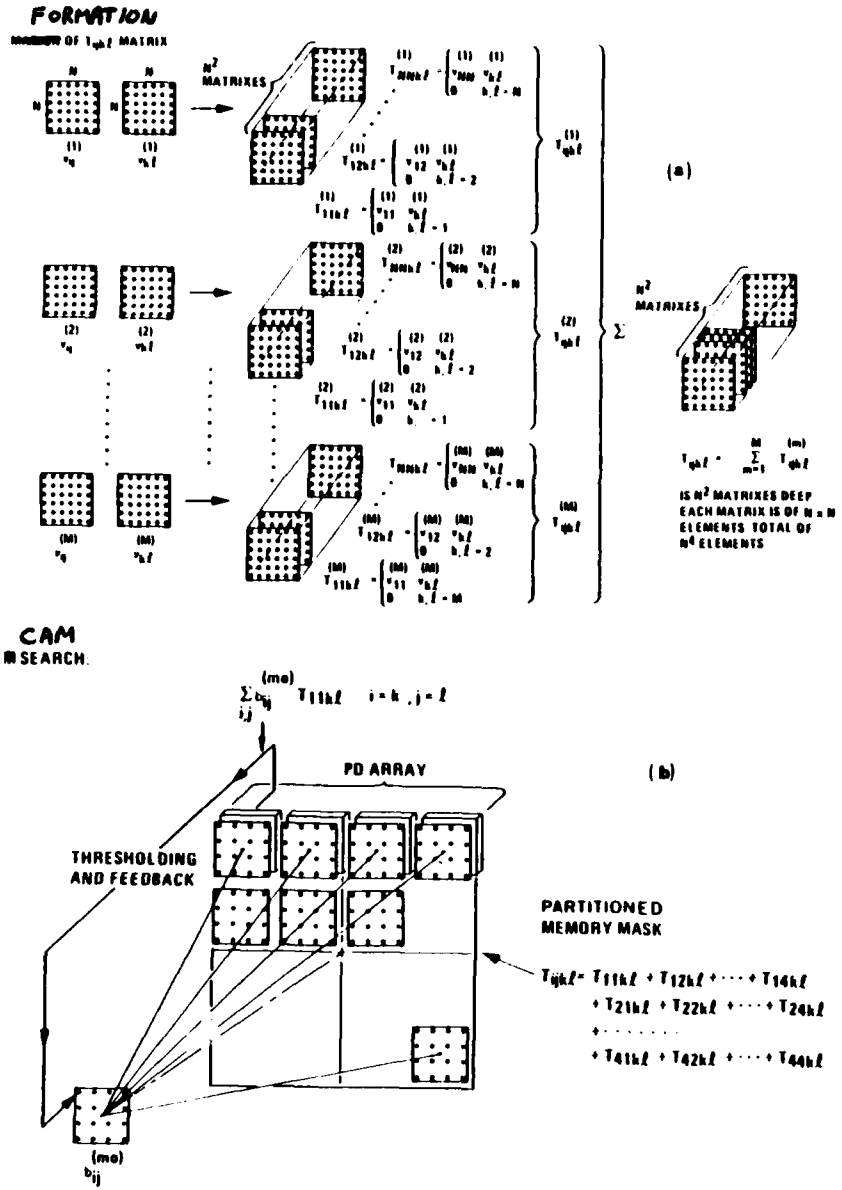


Fig. 12. (a) Formation of a four-dimensional connectivity matrix for a two-dimensional neural net and (b) memory search based on partitioning of the four-dimensional connectivity matrix.

This is followed by thresholding to obtain a new $N \times N$ matrix, which is used to replace $b_{kl}^{(m)(1)}$ in Eq. (14) for subsequent iterations. The procedure is repeated until the resulting matrix converges to the stored entity closest to the initiating matrix $b_{ij}^{(m)(0)}$.

The operation in Eq. (14) can be interpreted as first partitioning of the four-dimensional T_{ijkl} matrix into a two-dimensional array of two-dimensional submatrices: $T_{11kl}, T_{12kl}, \ldots, T_{1Nkl}, \ldots, T_{21kl}, T_{22kl}, \ldots, T_{2Nkl}, \ldots, T_{N1kl}, T_{N2kl}, \ldots, T_{NNkl}$ as depicted schematically in Fig. 12b where the partitioned submatrices are arranged in a two-dimensional array. This first step is followed by multiplication of $b_{kl}^{(m)(0)}$ by each of the partitioned submatrices, on an element-by-element basis and summing the products for each submatrix to obtain the first estimate $\hat{b}_{ij}^{(m)(1)}$. A numerical simulation of a two-dimensional neural network of 5×5 neurons based on the above algorithm and partitioning of the T_{ijkl} memory matrix is presented in Figs. 13-15. Three two-dimensional unipolar binary entities $b_{ij}^{(m)(1)}$, $m = 1, 2, 3$, are stored. These are shown in Fig. 13. The resulting T_{ijkl} matrix is given in Fig. 14 in partitioned form. The results of recall from partial versions (20%) of the three entities are presented in Fig. 15. These show that complete correct recall of each entity takes place after two iterations.

The summation operation called for in Eq. (14) is carried out in Fig. 12b by placing a spatially integrating photodetector behind each submatrix of the partitioned-memory mask which is assumed for the time being to be realized as pixel transmittance modulation in a hypothetical transparency capable of assuming negative transmittance values. The input entity $b_{ij}^{(m)(1)}$, for example in Fig. 12b, is assumed to be displayed on a suitable LED array. The LED display of $b_{ij}^{(m)(1)}$ is multiplied by the ideal transmittance of each partition submatrix by imaging the display on each of these with exact registration of pixels by means of a lenslet array as depicted in Fig. 16. The output of each photodetector, proportional to one of the components of Eq. 14, is thresholded, amplified, and fed back to drive an associated LED. The (i, j)th LED is paired with the (i, j)th photodetector. This completes the interconnection of the two-dimensional array of $N \times N$ neurons in the above architecture, where each neuron communicates its state to all other neurons through a prescribed four-dimensional connectivity matrix in which information about M, $N \times N$ matrices have been stored. The maximum number of two-dimensional entities that can be stored in this fashion is $M \simeq N^2/(8 \ln N)$, which follows directly from the storage capacity formula for the one-dimensional neural-net case by replacing N by N^2.

The added complexity associated with having to realize a bipolar transmittance in the partitioned T_{ijkl} memory mask of Fig. 12b can be avoided by using unipolar transmittance. This can lead, however, to some degradation in performance. A systematic numerical simulation study (Lee and

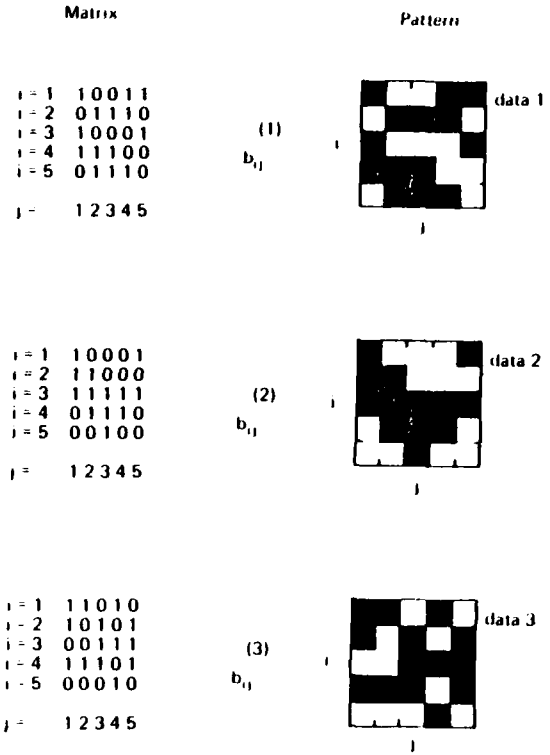


Fig. 13. Two-dimensional CAM simulation. Stored entities in matrix form (left) and in pictorial form (right).

Farhat, 1985) of a neural-net CAM in which statistical evaluation of the performance of the CAM for various types of memory masks (multivalued, clipped ternary, and clipped unipolar binary) and thresholding schemes (zero, threshold, adaptive threshold where the energy of the input vector is used as the threshold, adaptive thresholding and relaxation) was carried out. The results indicate that a unipolar binary memory mask can be used with virtually no sacrifice in CAM performance with the adaptive threshold and relaxation scheme. The scheme assumes an adaptive threshold proportional to the energy (light intensity) of the input entity displayed by the LED array at any time is used. The scheme of Fig. 12b can thus be realized by projecting an image of the input pattern directly onto an additional photodetection element. The detector output, being proportional to the total intensity of the input display, is used as a variable or adaptive threshold in a comparator against which the outputs of the PD elements positioned

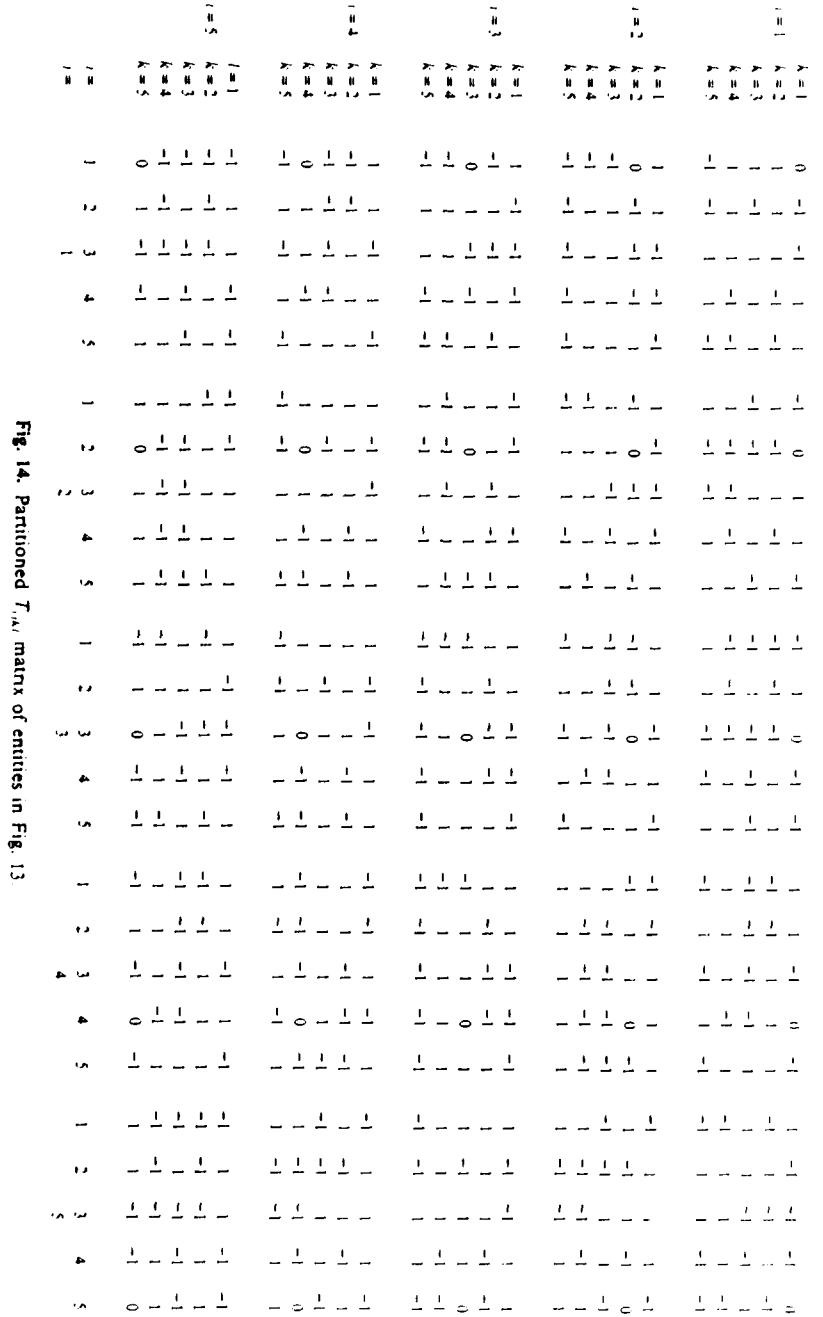


Fig. 14. Partitioned T_{ijkl} matrix of entities in Fig. 13.

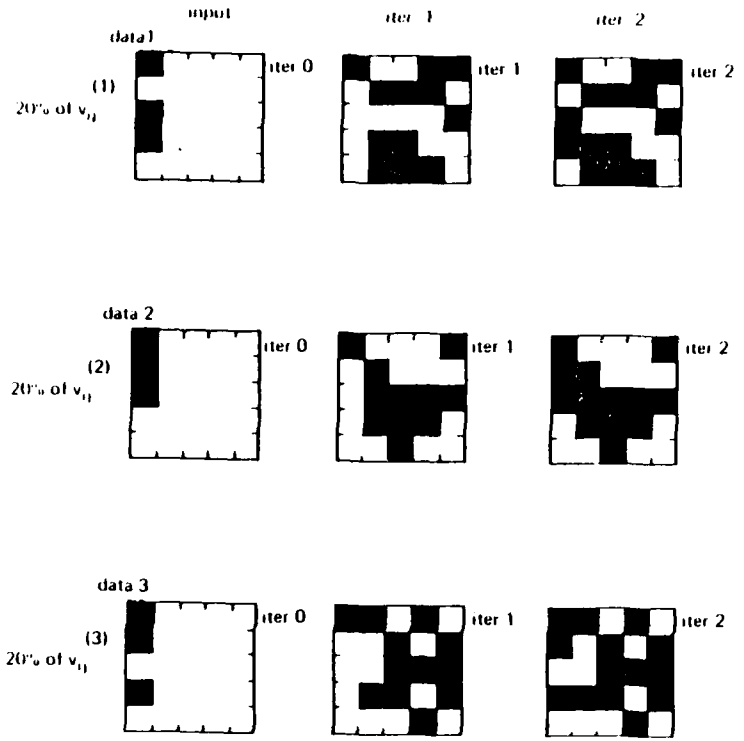


Fig. 15. Two-dimensional CAM simulation: Recognition of partial input.

behind the partitioned components of the T_{ijkl} *memory mask are compared* (subtracted). The comparator outputs are attenuated, and each is fed into a limiting amplifier with delayed feedback (relaxation). Each limiter-amplifier output is used to drive the LED that each photodetector is paired with. It is found (Lee and Farhat, 1985) that this scheme yields a performance equivalent to that of an ideal CAM with a multivalued connectivity matrix and zero thresholding. Note that although the initializing two-dimensional entities $b_{ij}^{(m_0)}$ are unipolar binary, the entity fed back after adaptive thresholding and limited amplification to drive the LED array are initially analog, resulting in multivalued iterates and intensity displays. However, after a sufficient number of iterations, the memory converges to the binary entities.

The ability to use unipolar binary memory matrices in the fashion described means that simple black-and-white photographic transparencies can be used as synaptic connectivity masks as suggested by the two schemes of Fig. 16. One scheme employs parallel optical feedback whereas the other

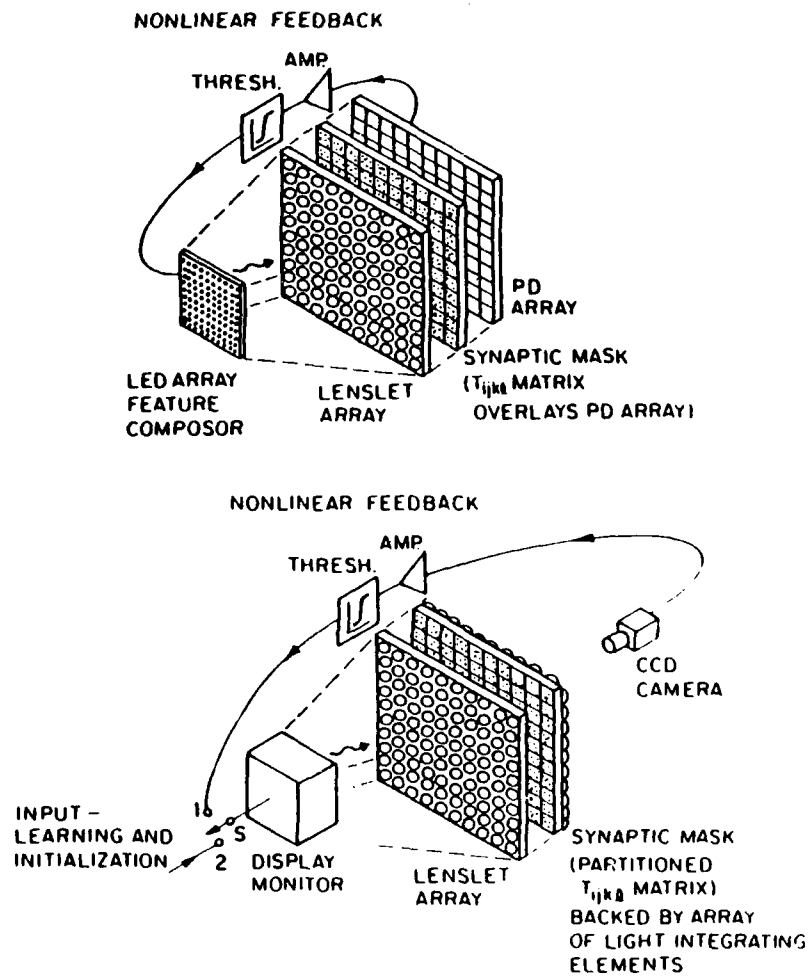


Fig. 16. Optoelectronic implementations of two-dimensional associative memory.

employs serial optoelectronic feedback. This latter scheme requires the use of a CCD television camera to enable time integration of the intensity distribution that is produced by the array of spatially integrating elements. These are diffusers situated behind each submask, and they randomly redistribute the light transmitted through each submask, thereby acting as integrating hemispheres. It is worth noting that this serial feedback scheme appears to be a novel means of achieving an arbitrary space-variant impulse

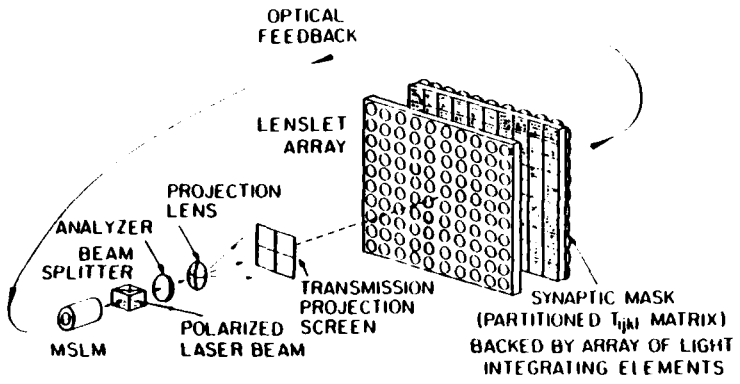


Fig. 17. Two-dimensional associative memory implemented with an electron-optic spatial light modulator.

response followed by a selected nonlinearity, which constitute two powerful operations in signal processing. A variation combining aspects of the above two schemes is shown in Fig. 17. In this arrangement, parallel nonlinear feedback is realized electron-optically with the aid of a microchannel spatial light modulator (MSLM) (Warde *et al.*, 1981; Warde and Thackara, 1982, also Chapter 7.2, this volume; Hara *et al.*, 1985, 1986). This device also provides an electronically controlled variable thresholding capability in which all parts of an output image relayed by the device falling below a given electronically adjustable intensity or brightness level are suppressed. The use of an MSLM in optical associative elements and computer architectures has also been described by Fisher, Giles and Lee (1985). An added advantage of a unipolar binary mask is the simplification in using a computer-driven nonvolatile spatial light modulator such as the Litton Magnetooptic SLM (Ross *et al.*, 1983) to realize a programmable CAM whose content can be altered, thus expanding its search capabilities using an electronic memory or a page-oriented holographic memory.

V. Holographic Associative Memories

Holography represents another link between optics and associative memories. It was recognized early (van Heerden, 1963; Gabor, 1969) that when a hologram of an object is recorded with a coded reference, the object is reconstructed when the hologram is illuminated with the same reference beam. Moreover, the reference beam can also be reconstructed by illuminating the hologram with the object beam. We can therefore think of the reference and object beams as being associated with each other. This is shown schematically in Fig. 18. The distinction in this case between an

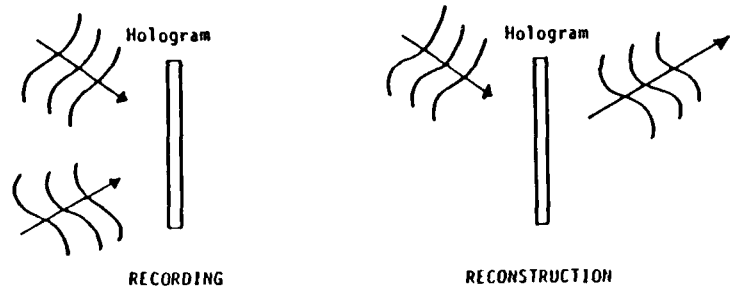


Fig. 18. Holographic associative memory.

object and a reference becomes unimportant; we simply think of two patterns being associated with each other. Distorted partial versions of either pattern can also reconstruct the other, and if a portion of the recorded hologram is eliminated, the recorded pattern can be reconstructed with fidelity that degrades gradually as the portion of the hologram that is removed increases. These properties are reminiscent of the behavior exhibited by the outer-product associative memories discussed earlier. However, there is a crucial difference between holographic associative memories and the outer-product memories that limits our ability to store more than one pair of associated patterns on the same hologram.

To appreciate this distinction we consider a Fresnel hologram on which two separate pairs of associations are simultaneously recorded. Let f_1, f_2 and g_1, g_2 denote the two pairs and let the same symbols primed represent the Fresnel diffraction of the original patterns. The amplitude transmittance of the hologram can then be written as

$$T(x, y) = f_1'(x, y)f_2'^*(x, y) + g_1'(x, y)g_2'^*(x, y) + \text{bias terms} \qquad (15)$$

Now imagine this hologram being illuminated with a light beam that is modulated by the pattern $f_2(x, y)$, positioned with respect to the hologram in precisely the same manner as it was during recording. Then the light incident on the hologram is $f_2'(x, y)$, and the two terms of interest in the diffracted light are

$$f_1'(x, y)|f_2'(x, y)|^2 + g_1'(x, y)g_2'^*(x, y)f_2'(x, y) \qquad (16)$$

where $|f_2'(x, y)|^2$ is recognized as the intensity of the light diffracted from the object f_2. If the distance between f_2 and the hologram is sufficiently long, the intensity of the diffracted light will have very smooth variations. Consequently the first term of the reconstruction in Eq. (16) can be approximated by $f_1'(x, y)$. This diffracted light can be transformed with the aid of a lens to reproduce an image of f_1. If there is no special relationship between f_2 and g_2, the second term in Eq. (16) produces a highly convolved and

unrecognizable image. In general the observed pattern due to this second term has a random, speckle like appearance, and its effect is to cause interference when we try to view the reconstruction of f_1. This behavior is also similar to that of the outer-product memory in that the "correct" association is obtained along with interfering cross terms. The difference is that the cross terms in the outer-product memory are suppressed by the square root of the space-bandwidth product of the input patterns. This does not happen in a hologram. In Eq. (16) the cross-product term is as strong as the first (signal) term. The only difference is that the reconstruction of the first term is recognizable as the pattern f_1 whereas the second term is noiselike. If we store two arbitrary images on the same hologram and reconstruct the hologram by illuminating it with one of the stored patterns, then on the average the signal-to-noise ratio of the reconstruction will be unity.

This limitation of holographic associative memories was perhaps recognized by Gabor who never mentioned the possibility of recording multiple associations on a single Fresnel hologram. It is, however, possible to record multiple associations using holography with one of two techniques: volume holography and multiplexing on planar holograms.

Sensitivity to the angle of illumination is a property that distinguishes volume from planar holograms. This property can be used to suppress the cross terms that appear in planar holograms; thus it is possible to store multiple associations in a volume hologram. Gaylord and coworkers at Georgia Tech (Guest and Gaylord, 1980) have explored associative memories using volume holography and have applied these memories to the optical calculation of look-up tables for performing binary multiplication. Here we shall discuss how multiplexed planar holograms can be used to synthesize associative memories that can store two-dimensional patterns and are functionally equivalent to the outer-product memories discussed in earlier sections.

The optical system shown in Fig. 19 was recently demonstrated experimentally (Paek and Psaltis, 1986) and is a modification of a system presented earlier (Psaltis and Farhat, 1985). As described in the previous section, the linear transformation that must be performed when we attempt to store two-dimensional patterns in an outer-product associative memory involves a four-dimensional kernel that cannot be directly implemented optically. Replacing the discrete one-dimensional vectors $v_i^{(m)}$ with continuous two-dimensional functions $f^{(m)}(x, y)$, we obtain the following four-dimensional function, which is analogous to the T_{ijkl} obtained in Eq. (13):

$$T(x, y, \hat{x}, \hat{y}) = \sum_{m}^{M} f^{(m)}(x, y) f^{(m)}(\hat{x}, \hat{y}) \tag{17}$$

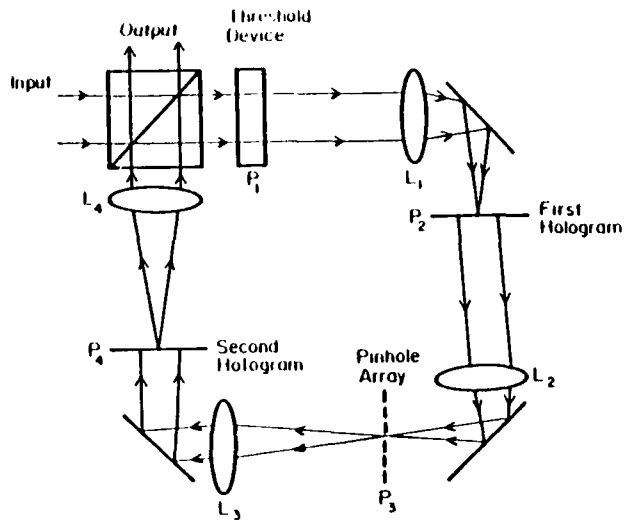


Fig. 19. Holographic implementation of a neural network model of associative memory

We obtain an expression analogous to Eq. (10) for the output produced by the two-dimensional outer-product memory when it is addressed with an input image $f(x, y)$:

$$\hat{f}(\hat{x}, \hat{y}) = \iint T(x, y, \hat{x}, \hat{y}) f(x, y)\, dx\, dy$$

$$= \sum_{m}^{M} \left(\iint f^{(m)}(x, y) f(x, y)\, dx\, dy \right) f^{(m)}(\hat{x}, \hat{y}) \tag{18}$$

The two-dimensional integration over x and y in this equation is equivalent to a two-dimensional correlation between the images $f(x, y)$ and $f^{(m)}(x, y)$ evaluated at the origin of the correlation plane. This observation suggests that a coherent optical correlator (VanderLugt, 1964) can be used to implement each of these inner products. The overall system is shown in Fig. 19. An input image enters the system through the beamsplitter. A spatial light modulator is used in plane P_1 to detect the image incident on it and produce on its other side a thresholded version of the incident image. Lens L_1 produces at plane P_2 the Fourier transform of the light amplitude at P_1. A Fourier transform hologram is placed at P_2 that contains all the stored references. The hologram is constructed as shown in Fig. 20. The images that are to be stored in the memory are arranged side by side in a composite large transparency. A Fourier transform hologram of the composite is then

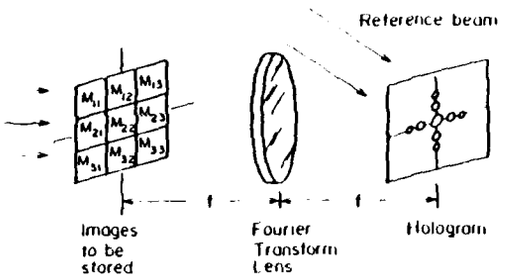


Fig. 20. Recording of the Fourier transform holograms used in the implementation of Fig. 19.

formed. Lens L_2 in Fig. 19 is also a Fourier transforming lens, which results in the light amplitude at plane P_3 being the correlation between the image at P_1 and the composite photograph. Due to the shift invariance of the optical correlator, the correlation patterns between the input and the individual images that make up the composite are formed spatially separated at plane P_3. Since we must keep only the inner products, not the full correlations (Psaltis *et al.*, 1986), an array of pinholes is placed at plane P_3 to sample each correlation pattern at its origin. Thus the amplitude of the light transmitted through each pinhole is approximately proportional to

$$\hat{f}^{(m)} = \iint f^{(m)}(x - \hat{x}, y - \hat{y}) f(x, y)\, dx\, dy \Big|_{\hat{x}=0,\hat{y}=0}$$

$$= \iint f^{(m)}(x, y) f(x, y)\, dx\, dy \tag{19}$$

The light transmitted by each pinhole is collimated by lens L_3 and illuminates the second Fourier transform hologram placed at plane P_4. This second hologram is fabricated in the same way as the first and contains the same images. The final lens L_4 produces the Fourier transform of the light at plane P_4 back at plane P_1. Thus the optical system from P_4 to P_1 is also a correlator. If we approximate each pinhole by a delta function, then the light transmitted through each pinhole reconstructs the entire composite image stored in the hologram. The light transmitted though a pinhole on which the mth inner product is formed is located at plane P_1 at the position of the mth image in the composite photograph. As a result the reconstruction obtained at P_1 due to this pinhole is shifted such that the mth image appears centered around the optical axis. Therefore the light incident on plane P_1 is a superposition of all the stored images, each weighted by the appropriate inner product precisely as prescribed by Eq. (18). The spatial light modulator thresholds this light distribution, and the result becomes the input for the next iteration around the optical loop. When the system has reached stability, the result can be probed through the beam splitter as shown in Fig. 19.

VI. Conclusion

The implementation of optical associative memories described in this chapter establishes a link between neural network modeling and optics that we hope will prove useful in the implementation of optical information processing systems. A basic compatibility exists between what optics has to offer and what is required for the simulation of neural network models. As new models emerge and their sophistication increases, we can expect that optical implementations of these models will continue to show advantages over other approaches. Therefore neural networks and related models of computation may provide the appropriate algorithmic framework that will guide the development of optical information processing systems.

Acknowledgments

We acknowledge our colleagues Y. S. Abu-Mostafa, J. Hong, E. G. Paek, A. Prata, S. Venkatesh, and K. S. Lee with whom we have collaborated in several aspects of the work in this area.

This research is funded by DARPA as a joint project at Caltech and the University of Pennsylvania.

References

Amari, S. (1977). *Biol. Cybern.* **26**, 175–185.
Andersen, D. Z. (1986). *Opt. Lett.* **11**, 56–58.
Anderson, J. A., Silverstein, J. W., Ritz, S. A., and Jones, R. S. (1977). *Psychol. Rev.* **34**, 413–451.
Athale, R. A., Friedlander, C. B., and Kushner, B. G. (1986). *SPIE Semin. Proc.* **625**, 179.
Cohen, M. (1986). *SPIE Semin. Proc.* **625**, 215.
Condon, D. J., Reichenbach, M. C., Tarasevich, A., and Rhodes, W. T. (1985). *OSA Top. Meet. Opt. Comput., Incline Village, Nev.* Post-deadline pap., PD-2.
Farhat, N., and Miyahara, S. (1986). *Tech. Digest, Spring OSA Top. Meet. Signal Recovery Synth. II*, Hono, Hawaii, pp. 120–123.
Farhat, N., and Psaltis, D. (1985). *Tech. Digest, OSA Annu. Meet., Washington, D.C.* p. 58.
Farhat, N., Psaltis, D., Prata, A., and Paek, E. (1985). *Appl. Opt.* **24**, 1469–1475.
Fisher, A. D., Giles, C. L., and Lee, J. N. (1985). *Tech. Dig., OSA Top. Meet. Opt. Comput.* pp. WB4-1–WB4-4.
Gabor, D. (1969). *IBM J. Res. Dev.* **13**, 156–159.
Gibbs, H. M., McCall, S. L., and Venkatesan, T. N. C. (1981). *SPIE Semin. Proc.* **269**, 75–80.
Goodman, J., Dias, R. A., and Woody, L. M. (1978). *Opt. Lett.* **2**, 1.
Graph, H. P. (1985). *Workshop Neural Networks Comput., Santa Barbara, California.*
Grossberg, S. (1982). "Studies of Mind and Brain." Reidel, Dordrecht and Boston.
Guest, C., and Gaylord, T. (1980). *Appl. Opt.* **19**, 1201.
Hara, T., Sugiyama, M., and Suzuki, Y. (1985). *Adv. Electron. Electron. Devices* **64B**, 637–647.
Hara, T., Ooi, Y., Kato, T., and Suzuki, Y. (1986). *SPIE Semin. Proc.* **613**, 25.
Hebb, D. (1949). "Organization of Behavior." Wiley, New York.

Hecht-Nielsen, R. (1983). *SPIE Semin. Proc.* **360**, 180–189.
Hopfield, J. J. (1982). *Proc. Nat. Acad. Sci. USA* **79**, 2554–2558.
Kohonen, T. (1972). *IEEE Trans. Comput.* **C-21**, 353–359.
Kohonen, T. (1978). "Associative Memory." Springer-Verlag, Berlin and New York.
Kohonen, T. (1984). "Self-Organization and Associative Memory." Springer-Verlag, Berlin and New York.
Lambe, J., Thakoor, A. P., and Moopen, A. (1985). *JPL Rep.* D 2825.
Lee, K. S., and Farhat, N. (1985). *Tech. Dig. OSA Annu. Meet., Washington, D.C.* p. 48.
Little, W. A. (1974). *Math. Biosci.* **19**, 101–120.
Longuet-Higgins, H. C., Willshaw, D. J., and Buneman, O. P. (1970). *Q. Rev. Biophys.* **3**, 223–224.
McEliece, R. J., Posner, E. C., Rodemich, E. R., and Venkatesh, S. (1986). Caltech Rep. Also, *IEEE Trans. Inf. Theory* (submitted).
Miller, D. A. B., Smith, D., and Colin, S. (1981). *IEEE J. Quant. Electron.* **QE-17**, 312–317.
Nakano, K. (1972). *IEEE Trans. Syst. Man Cybern.* **SMC-2**, 380–388.
Paek, E. G., and Psaltis, D. (1986). *Annu. Meet. Opt. Soc. Am., Seattle, Wash.* Abstr. FMDD6.
Poggio, T., and Koch, C. (1985). *Proc. R. Soc. London, Ser. B* **226**, 303–323.
Porada, Z. (1983). *Thin Solid Films* **109**, 213–216.
Psaltis, D., and Farhat, N. (1985). *Opt. Lett.* **10**, 98–100.
Psaltis, D., Hong, J., and Venkatesh, S. D. (1986). *SPIE Semin. Proc.* **625**, 189.
Ross, W., Psaltis, D., and Anderson, R. (1983). *Opt. Eng.* **22**, 485–490.
Soffer, B., Dunning, G. J., Owechko, Y., and Marom, E. (1986). *Opt. Lett.* **11**, 118–120.
Tanguay, A. R., and Warde, C. (Eds.) (1983). Special issue on SLM's: Critical Issues, *Opt. Eng.* **22**, 663.
Tikhonov, A. N., and Arsenin, V. Y. (1977). "Solutions of Ill-Posed Problems." Winston, Washington, D.C.
VanderLugt, A. (1964). *IEEE Trans. Inf. Theory* **IT-10**, 139.
van Heerden, P. J. (1963). *Appl. Opt.* **2**, 387, 393.
Venkatesh, S., and Psaltis, D. (1985). *Workshop Neural Networks Comput., Santa Barbara, California.* Also, *IEEE Trans. Inf. Theory* (submitted), 1985.
Warde, C., and Thackara, J. I. (1982). *Opt. Lett.* **7**, 244–246.
Warde, C., Weiss, A. M., Fisher, A. D., and Thackara, J. I. (1981). *Appl. Opt.* **20**, 2066–2074.
Willshaw, D. J. (1971). Doctoral Dissertation, Edinburgh University, Scotland.
Willshaw, D. J. (1972). *Proc. R. Soc. London, Ser. B* **182**, 253–257.
Willshaw, D. J., Buneman, O. P., and Longuet-Higgins, H. C. (1969). *Nature (London)* **222**, 960–962.
Yariv, A., and Kwong, S. K. (1986). *Opt. Lett.* **11**, 183–186.

Reprinted from Optics Letters, Vol. 12, page 448, June 1987.

Architectures for optoelectronic analogs of self-organizing neural networks

Nabil H. Farhat

Department of Electrical Engineering, Electro-Optics and Microwave-Optics Laboratory, The University of Pennsylvania, Philadelphia, Pennsylvania 19104-6390

Received December 12, 1986, accepted March 20, 1987

Architectures for partitioning optoelectronic analogs of neural nets into input–output and internal groups to form a multilayered net capable of self-organization, self-programming, and learning are described. The architectures and implementation ideas given describe a class of optoelectronic neural net modules that, when interfaced to a conventional computer controller, can impart to it artificial intelligence attributes.

In earlier work on optical analogs of neural nets,[1–6] the nets described were programmed to do a specific computational task, namely, a nearest-neighbor search consisting of finding the stored entity that is closest to the address in the Hamming sense. The net acted as a content-addressable associative memory. The programming was done by first computing the interconnectivity matrix using an outer-product recipe given the entities that one wished the net to store and then setting the weights of synaptic interconnections between neurons accordingly.

In this Letter we are concerned with architectures for optoelectronic implementation of neural nets that are able to program or organize themselves under supervised conditions, i.e., of nets that are capable of (1) computing the interconnectivity matrix for the associations that they are to learn and (2) changing the weights of the links between their neurons accordingly. Such self-organizing networks therefore have the ability to form and store their own internal representations of the associations that they are presented with.

Multilayered self-programming nets were described as early as 1969,[7] and in more recent descriptions[8–10] the net is partitioned into three groups. Two are input and output groups of neurons that interface with the net environment. The third is a group of hidden or internal units that separates the input and output units and participates in the process of forming internal representations of the associations that the net is presented with.

Two supervised learning procedures in such partitioned nets have recently attracted attention. One is stochastic, involving a simulated annealing process,[11,12] and the other is deterministic, involving an error backpropagation process.[9] There is general agreement, however, that because of their iterative nature, sequential computation of the links using these algorithms is time consuming. A faster means for carrying out the required computations is needed.

Optics and optoelectronic architectures and techniques can play an important role in the study and implementation of self-programming networks and in speeding up the execution of learning algorithms. Here we describe a method for partitioning an optoelectronic analog of a neural net to implement a multilayered net analog that can learn stochastically by means of a simulated annealing learning algorithm in the context of a Boltzmann machine formalism [see Fig. 1(a)]. The arrangement shown in Fig. 1(a) derives from the neural network analogs that we described earlier.[2] The network, consisting of, say, N neurons, is partitioned into three groups. Two groups, V_1 and V_2, represent input and output units, respectively. The third group, H, comprises hidden or internal units. The partition is such that $N_1 + N_2 + N_3 = N$, where N_1, N_2, and N_3 refer to the number of neurons in the V_1, V_2, and H groups, respectively. The interconnectivity matrix, designated here W_{ij}, is partitioned into nine submatrices, A–F, and three zero submatrices, shown as blackened or opaque regions of the W_{ij} mask. The LED array represents the state of the neurons, assumed to be unipolar binary (LED on, neuron firing; LED off, neuron not firing). The W_{ij} mask represents the strengths of interconnection among neurons in a manner similar to earlier arrangements.[2] Light from each LED is smeared vertically over the corresponding column of the W_{ij} mask with the aid of an anamorphic lens system [not shown in Fig. 1(a)], and light emerging from each row of the mask is focused with the aid of another anamorphic lens system (also not shown) onto the corresponding elements of the photodetector (PD) array. The scheme utilized in Ref. 2 for realizing bipolar values of W_{ij} in incoherent light is adopted here; it consists of separating each row of the W_{ij} mask into two subrows, assigning positive-valued W_{ij} to one subrow and negative-valued W_{ij} to the other, and focusing light emerging from the two subrows separately onto two adjacent photosites connected in opposition in the photodetector array. Submatrix A, with $N_1 \times N_1$ elements, provides the interconnection weights between units or neurons within group V_1. Submatrix B, with $N_2 \times N_2$ elements, provides the interconnection weights between units within V_2. Submatrices C (with $N_1 \times N_3$ elements) and D (with $N_3 \times N_1$ elements) provide the interconnection weights between units of V_1 and H, and submatrices E (with $N_2 \times N_3$ elements) and F

0146-9592/87/060448-03$2.00/0

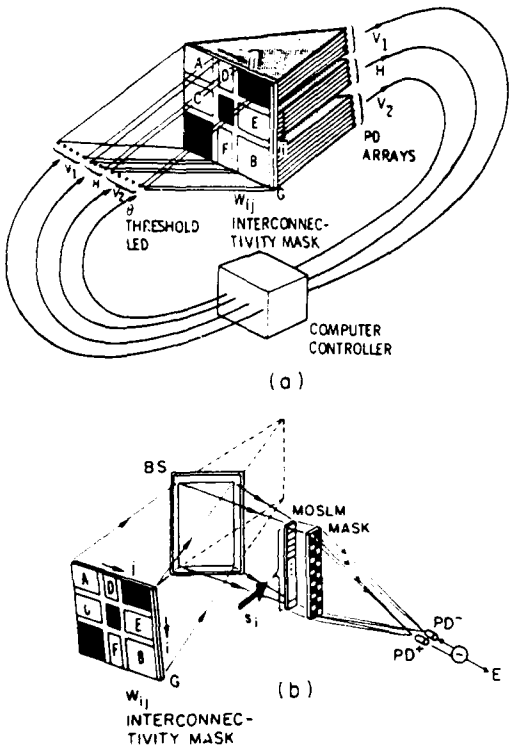


Fig. 1. Architecture for optoelectronic analog of layered self-programming net. (a) Partitioning concept showing adjustable global threshold scheme. (b) Arrangement for rapid determination of the net's global energy E for use in learning by simulated annealing.

(with $N_1 \times N_2$ elements) provide the interconnection weights of units of V_2 and H. Units in V_1 and V_2 cannot communicate with one another directly because the locations of their interconnectivity weights in the W_{ij} matrix or mask are blocked out (blackened lower-left and top-right portions of W_{ij}). Similarly, units within H do not communicate with one another because locations of their interconnectivity weights in the W_{ij} mask are also blocked out (center blackened square of W_{ij}). The LED element θ is of graded response. Its output represents the state of an auxiliary neuron in the net that is always on to provide a global threshold level to all units by contributing only to the light focused onto negative photosites of the PD arrays from pixels in the G column of the interconnectivity mask. This is achieved by suitable modulation of pixels in the G column. This method for introducing the threshold level is attractive, as it allows for providing to all neurons in the net a fixed global threshold, an adaptive global threshold, or even a noisy global threshold.

By using a computer-controlled nonvolatile spatial light modulator to implement the W_{ij} mask in Fig. 1(a) and including a computer–controller as shown, the scheme can be made self-programming with the ability to modify the weights of synaptic links between its neurons. This is done by fixing or clamping the states of the V_1 (input) and V_2 (output) groups to each of the associations that we want the net to learn and by repeated application of the simulated annealing procedure with the Boltzmann or another stochastic state-update rule and collecting statistics on the states of the neurons at the end of each run when the net reaches thermodynamic equilibrium.

Starting from an arbitrary W_{ij}, and for each clamping of the V_1 and V_2 units to one of the associations, the states of units in H are switched and annealing is applied, until thermodynamic equilibrium is reached. The state vector of the entire net, which represents a state of the global energy minimum, is then stored by the computer. This procedure is repeated for each association several times and the final state vectors recorded every time. Note that, because of the probabilistic nature of the state-update rule discussed later and in Eqs. (1) and (2) below, the states of global energy minimum in the runs for each association may not necessarily be exactly the same. Therefore the need to collect statistics from which the probabilities p_{ij} of finding the ith and jth neurons in the same state can then be obtained. Next, with the output units V_2 unclamped to let them run free, the above procedure is repeated for the same number of annealings as before and the probabilities p_{ij}' are obtained. The weights W_{ij} are then incremented by $\Delta W_{ij} = \eta(p_{ij} - p_{ij}')$, where η is a constant that controls the speed and efficacy of learning. Starting from the new W_{ij}, the above procedure is repeated until a steady state W_{ij} is reached, at which stage the learning procedure is complete.

Learning by simulated annealing requires calculating the energy, E, of the net[8,10]:

$$E = -\frac{1}{2}\sum_{i} u_i s_i, \tag{1}$$

where s_i is the state of the ith neuron and

$$u_i = \sum_{j \neq i} W_{ij} s_j - \theta_i + I_i, \tag{2}$$

respectively. A simplified version of a rapid scheme for obtaining E optoelectronically is shown in Fig. 1(b). The scheme requires the use of an electronically addressed nonvolatile binary (on–off) spatial light modulator (SLM) consisting of a single column of N pixels. A suitable candidate is a parallel-addressed magneto-optic SLM (MOSLM) consisting of a single column of N pixels that are driven electronically by the same signal driving the LED array in order to represent the state vector $\mathbf{s}$ of the net. A fraction of the focused light emerging from each row of the W_{ij} mask is deflected by the beam splitter BS onto the individual pixels of the column MOSLM such that light from adjacent pairs of subrows falls upon one pixel of the MOSLM. The MOSLM pixels are overlaid by a checkered binary mask as shown. The opaque and transparent pixels in the checkered mask are staggered in such a fashion that light emerging from the left subcolumn will be derived from the positive subrows W_{ij}^{+} of W_{ij} and light emerging from the right subcolumn will be derived from the negative subrows W_{ij}^{-} of W_{ij}. By separately focusing the light from the left and right subcolumns as shown onto two photodetectors and subtracting and halving their outputs, one obtains

$$E = -\frac{1}{2}\sum_i \left[\left(\sum_{j\neq i} W_{ij}^{+} - W_{ij}^{-}\right)s_j\right]s_i$$

$$= -\frac{1}{2}\sum_i \left(\sum_{j\neq i} W_{ij}s_j\right)s_i, \qquad (3)$$

which is the required energy. In Eq. (3) the contributions of θ_i and I_i in Eq. (2) are absorbed in W_{ij}. The simulated annealing algorithm involves determining the change ΔE_k in E that is due to switching the state of the kth neuron in H selected at random, computing the probability $p(\Delta E_k) = 1/[1 + \exp(-\Delta E_k/T)]$, and comparing the result with a random number $n_r \epsilon (0, 1)$ produced by a fast random-number generator. If $p(\Delta E_k) > n_r$, the change in the state of the kth neuron is retained. Otherwise it is discarded and the neuron is returned to its original state before a new neuron in H is randomly selected and switched and the annealing procedure repeated. This algorithm ensures that in thermal equilibrium the relative probability of two global states follows a Boltzmann distribution[8]; hence it is sometimes referred to as the Boltzmann machine. In this fashion the search for a state of global energy minimum is done by employing a gradient-descent algorithm that allows for probabilistic hill climbing. The annealing process usually also includes a cooling schedule in which the temperature T is allowed to decrease between iterations to increase gradually the fineness of search for the global energy minimum. Optoelectronic methods for generating $p(\Delta E_k)$ that employ speckle statistics[13] and for generating the random number n_r by photon counting[14] can also be incorporated to speed up the procedure and reduce the number of digital computations involved.

The architecture described here for partitioning a neural net can be used in hardware implementation and study of self-programming and learning algorithms such as the simulated annealing algorithm outlined here. The parallelism and massive interconnectivity provided through the use of optics should markedly speed up learning even for the simulated annealing algorithm, which is known to be quite time consuming when carried out on a sequential machine. The partitioning concept described is also extendable to multilayered nets of more than three layers and to the two-dimensional arrangement of synaptic inputs to neurons, as opposed to the one-dimensional or lineal arrangement described here. Other learning algorithms calling for a multilayered architecture, such as the error backpropagation algorithm,[9,15] can also now be envisaged optoelectronically by employing the partitioning scheme described here or variations of it.

Learning algorithms in layered nets lead to analog or multivalued W_{ij}. Therefore high-speed computer-controlled SLM's with graded pixel response are called for. Methods of reducing the needed dynamic range of W_{ij} or for allowing the use of ternary W_{ij} are, however, under study to permit the use of commercially available fast nonvolatile binary SLM devices such as the Litton/Semetex MOSLM.[16] It is worth noting that the role of optics in the architecture described not only facilitates partitioning the net into groups or layers but also provides the massive interconnectivity mentioned earlier. For example, for a neural net with a total of $N = 512$ neurons, the optics permit making $2N^2 = 2.62 \times 10^5$ programmable weighted interconnections among the neurons in addition to the $4N = 2048$ interconnections that would be needed in the arrangement shown Fig. 1(b) to compute the energy E.

Assuming that material and device requirements of the architectures described can be met and partitioned, self-organizing neural net modules will be routinely constructed; then the addition of such a module to a computer–controller through a high-speed interface can be viewed as providing the computer–controller with artificial intelligence capabilities by imparting to it neural net attributes. These capabilities include self-organization, self-programmability and learning, and associative memory capability for conducting nearest-neighbor searches. Such attributes would enable a small computer to perform powerful computational tasks of the kind needed in pattern recognition and in the solution of combinatorial optimization problems and ill-posed problems encountered, for example, in inverse scattering and vision, which are confined at present to the domain of supercomputers.

The research reported was supported by grants from the Defense Advanced Research Projects Agency–Naval Research Laboratory, the U.S. Army Research Office, and The University of Pennsylvania's Laboratory for Research on the Structure of Matter.

References

1. D. Psaltis and N. Farhat, Opt. Lett. **10**, 98 (1985).
2. N. H. Farhat, D. Psaltis, A. Prata, and E. Paek, Appl Opt. **24**, 1469 (1985).
3. A. D. Fischer, C. Giles, and J. Lee, J. Opt. Soc. Am. A 1, 1337 (1984).
4. D. Z. Andersen, Opt. Lett. **11**, 56 (1986).
5. B. Soffer, G. Dunning, Y. Owechko, and E. Marom, Opt. Lett. **11**, 118 (1986).
6. A. Yariv and S. K. Kwong, Opt. Lett. **11**, 186 (1986).
7. S. Grossberg, J. Statist. Phys. 1, 319 (1969).
8. D. H. Ackley, G. Hinton, and T. Sejnowski, Cognit. Sci. 1, 147 (1985).
9. D. E. Rumelhart and all authors, in *Parallel Distributed Processing*, D. F. Rumelhart and J. L. McClelland, eds. (Bradford/MIT, Cambridge, Mass., 1986), Vol. 1.
10. T. J. Sejnowski and C. R. Rosenberg, Electrical Engineering and Computer Science Tech. Rep. JHU/EECS-96/01 Johns Hopkins University, Baltimore, Md., 1986).
11. N. Metropolis, A. Rosenbluth, M. Rosenbluth, and A. Teller, J. Chem. Phys. **21**, 1087 (1953).
12. S. Kirkpatric, C. Gelatt, Jr., and M. Vecchi, Science **220**, 671 (1983).
13. A. Ticknor, H. Barrett, and R. Easton, Jr., in *Digest of Topical Meeting on Optical Computing* (Optical Society of America, Washington, D.C., 1985), postdeadline paper PD-4.
14. J. Marron, A. Martino, and G. Morris, Appl. Opt. **25**, 26 (1986).
15. K. Wagner and D. Psaltis, Proc. Soc. Photo-Opt. Instrum. Eng. **756** (to be published).
16. W. Ross, Opt. Eng. **22**, 485 (1983).

Appendix III

Reprinted from Applied Optics, Vol. 26, page 5093, December 1, 1987

Optoelectronic analogs of self-programming neural nets: architecture and methodologies for implementing fast stochastic learning by simulated annealing

Nabil H. Farhat

Self-organization and learning is a distinctive feature of neural nets and processors that sets them apart from conventional approaches to signal processing. It leads to self-programmability which alleviates the problem of programming complexity in artificial neural nets. In this paper architectures for partitioning an optoelectronic analog of a neural net into distinct layers with prescribed interconnectivity pattern to enable stochastic learning by simulated annealing in the context of a Boltzmann machine are presented. Stochastic learning is of interest because of its relevance to the role of noise in biological neural nets. Practical considerations and methodologies for appreciably accelerating stochastic learning in such a multilayered net are described. These include the use of parallel optical computing of the global energy of the net, the use of fast nonvolatile programmable spatial light modulators to realize fast plasticity, optical generation of random number arrays, and an adaptive noisy thresholding scheme that also makes stochastic learning more biologically plausible. The findings reported predict optoelectronic chips that can be used in the realization of optical learning machines.

I. Introduction

Interest in neural net models (see, for example, Refs. 1–9) and their optical analogs (see, for example, Refs. 10–25) stems from well-recognized information processing capabilities of the brain and the fit between what optics can do and what even simplfied models of neural nets can offer toward the development of new approaches to collective signal processing.

Neural net models and their analogs present a new approach to collective signal processing that is robust, fault tolerant, and can be extremely fast. Collective or distributed processing describes the transfer among groups of simple processing units (e.g., neurons), that communicate among each other, of information that one unit alone cannot pass to another. These properties stem directly from the massive interconnectivity of neurons (the decision-making elements) in the brain and their ability to store information as weights of links between them, i.e., their synaptic interconnections, in a distributed nonlocalized manner. As a result, signal processing tasks such as nearest-neighbor searches in associative memory can be performed in time durations equal to a few time constants of the decision-making elements, the neurons, of the net. The switching time constant of a biological neuron is of the order of a few milliseconds. Artificial neurons (electronics or optoelectronic decision-making elements) can be made to be a thousand to a million times faster. Artificial neural nets can therefore be expected to function, for example, as content-addressable associative memory or to perform complex computational tasks such as combinatorial optimization which are encountered in computational vision, imaging, inverse scattering, superresolution, and automated recognition from partial, (sketchy) information, extremely fast in a time scale that is way out of reach for even the most powerful serial computer. In fact once a neural net is programmed to do a given task it will do it almost instantaneously. More about this point later. As a result optoelectronic analogs and implementations of neural nets are attracting considerable attention. Because of the noninteracting nature of photons, the optics in these implementations provide the needed parallelism and massive interconnectivity and therefore a potential for realizing relatively large neural nets while the decision-making elements are realized electronically heralding a possible ultimate marriage between VLSI and optics.

Architectures suitable for use in the implementation of optoelectronic neural nets of 1-D and 2-D arrangements of neurons were studied and described earlier.[10–15] Two-dimensional architectures for optoelectronic analogs have been successfully utilized in the

The author is with University of Pennsylvania, Electrical Engineering Department, Philadelphia, Pennsylvania 19104-6390.
Received 15 May 1987.
0003-6935/87/235093-11$02.00/0.

recognition of objects from partial information by either complementing the missing information or by automatically generating correct labels of the data (object feature spaces) the memory is presented with.[23] These architectures are based primarily on the use of incoherent light to help maintain robustness, by avoiding speckle noise and the strict positioning requirements encountered when use of coherent light is contemplated.

In associative memory applications, the strengths of interconnections between the neurons of the net are determined by the entities one wishes to store. Ideal storage and recall occurs when the stored vectors are randomly chosen, i.e., uncorrelated. Specific storage recipes based on a Hebbian model of learning (outer-product storage algorithm), or variations thereof, are usually used to explicitly calculate the weights of interconnections which are set accordingly. This represents explicit programming of the net, i.e., the net is explicitly taught what it should know. What is most intriguing, however, is that neural net analogs can also be made to be self-organizing and learning, i.e., become self-programming. The combination of neural net modeling, Boltzmann machines, and simulated annealing concepts with high-speed optoelectronic implementations promises to produce high-speed artificial neural net processors with stochastic rather than deterministic rules for decision making and state update. Such nets can form their own internal representations (connectivity weights) of their environment (the outside world data they are presented with) in a manner analogous to the way the brain forms its own representations of reality. This is quite intriguing and has far-reaching implications for smart sensing and recognition, thinking machines, and artificial intelligence as a whole. Our exploratory work is showing that optics can also play a role in the implementation and speeding up of learning procedures such as simulated annealing in the context of Boltzmann machine formalism,[26–29,49] and error backpropagation[30] in such self-teaching nets and for their subsequent use in automated robust recognition of entities the nets have had a chance to learn earlier by repeated exposure to them when the net is in a learning mode. Induced self-organization and learning seem to be what sets apart optical and optoelectronic architectures and processing based on models of neural nets from other conventional approaches to optical processing and have the advantage of avoiding explicit programming of the net which can be time-consuming and has come to be referred to as the programming complexity of neural nets.[48] The partitioning scheme presented in Sec. III permits defining input, output, and intermediate layers of neurons and any prescribed communication pattern between them. This enables the implementation of deterministic learning algorithms such as error backprojection. However, the discussion in this paper focuses on stochastic learning by simulated annealing since such learning algorithms may prove to be more biologically plausible since they might account for the noise present in biological neural nets as will be elaborated on in Sec. IV.

In this paper we are therefore concerned with architectures for optoelectronic implementation of neural nets that are able to program or organize themselves under supervised conditions, i.e., of nets that are capable of (a) computing the interconnectivity matrix for the associations they are to learn, and (b) changing the weights of the links between their neurons accordingly. Such self-organizing networks have therefore the ability to form and store their own internal representations of the entities or associations they are presented with. In Sec. II we attempt to elucidate those features that set neural processing apart from conventional approaches to signal processing. The ideas expressed have been arrived at as a result of maintaining a critical attitude and constantly keeping in mind, when engaged in the study of neural net models and their applications, the question of what is unique about the way they perform signal processing tasks. If they seem to perform a signal processing function well, could the same function be carried out equally well with a conventional processing scheme? To gain insight into this question we were led to a comparison between outer-product and inner-product schemes for implementing associative memory. The insight gained from this exercise points clearly to certain distinction between neural and conventional approaches to signal processing which will lead us to considerations of self-programmability and learning. These are presented in Sec. II together with a description of architectures for optoelectronic analogs of such self-organizing nets. The emphasis is on stochastic supervised learning, rather than deterministic learning, and on the use of noise to ensure that the combinatorial search procedure for a global energy or cost function during the learning phase does not get trapped in a local minimum of the cost function. In Sec. III a discussion of practical considerations related to the implementation of the architectures described and for accelerating the learning process is presented. An estimate of the speedup factor compared to serial implementation is included. Conclusions and implications of the work are then given. These attest to a continuing role for optics in the implementation of artificial neural net modules or neural chips with self-programming and learning capabilities, i.e., to optical learning machines.

II. Distinctive Features of Neural Processing

Right from the outset, when attention was first drawn to the fit between optics and neural models,[10,11] our investigations of optoelectronic analogs of neural nets and their applications have perpetually kept in view the question of what is it that neural nets can do that is not doable by conventional means, i.e., by well-established approaches to signal processing. Such critical attitude is found useful and almost mandatory to avoid being swept into ill-conceived research endeavors. It is not easy of course to see all the ramifications of a problem while one is immersed in its study and solution, but a critical attitude always helps to isolate real attributes from biased ones.

Being collective, adaptive, iterative, and highly non-

linear, neural net models and their analogs exhibit complex and rich behavior in their phase space or state space that is described in terms of attractors, limit points, and limit cycles with associated basins of attraction, bifurcation, and chaotic behavior. The rich behavior offers intellectually attractive and challenging areas of research. Moreover, many believe that in studying neural nets and their models we are attempting to benefit from nature's experience in its having arrived over a prolonged period of time, through a process of trial and error and retainment of those permutations that enhance the survivability of the organism, at a powerful, robust, and highly fault-tolerant processor, the brain, that can serve as the model for a new generation of computing machines. Clues and insights gained from its study can be immensely beneficial for use in artificially intelligent man-made machines that, like the brain, are highly suited for processing of spatiotemporal multi-sensory data and for motor control in a highly adaptive and interactive environment.

All the above are general attributes and observations that by themselves are sufficient justification for the interest displayed in neural nets as a new approach to signal processing and computation. To gain, however, further specific insight in what sets neural nets apart from other approaches to signal processing, we consider a specific example. This involves comparison between two mathematically equivalent representations of a neural net, one involving outer products, and the other inner products.[31] We begin by considering the optoelectronic neural net analog described earlier[12] and represented here in Fig. 1. The iterative procedure determining the evolution of the state vector $\bar{v}$ of the net is illustrated in Fig. 1(a) and the vector–matrix multiplication scheme with thresholding and feedback used to interconnect all neurons with each other through weights set by the T_{ij} mask is shown in Fig. 1(b). For a net of size N with interconnectivity matrix T_{ij}, where $T_{ii} = 0, i,j = 1,2 \ldots N$, the iterative equation for the state vector is

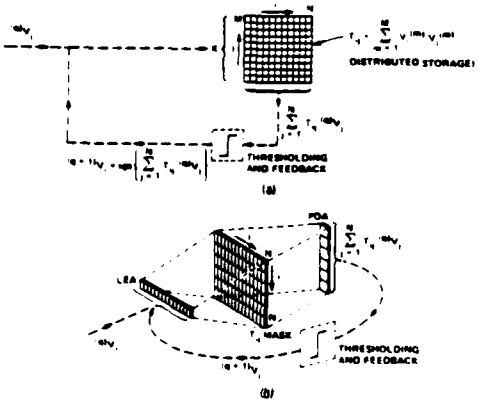


Fig. 1. Outer-product (distributed) storage and recall scheme.

$$ {}^{(q+1)}v_i = \operatorname{sgn}\left\{\sum_{j=1}^{N} T_{ij}\,{}^{(q)}v_j\right\}, \tag{1} $$

where the superscripts (q) and $(q+1)$ designate two consecutive iterations and sgn$\{\cdot\}$ represents the sign of the bracketed quantity. The iteration triggered by an externally applied initializing or strobing vector ${}^{(q)}\bar{v}$, $q = 0$, i.e., ${}^{(0)}\bar{v}$, continues until a steady-state vector that is one of the nominal state vectors or attractors of the net that is closest to ${}^{(0)}\bar{v}$ in the Hamming sense is converged upon. At this point the net has completed a nearest-neighbor search operation. For simplicity the usual terms for the threshold θ_i and external input I_i of the ith neuron have been omitted from Eq. (1). These can, without loss of generality of the conclusions arrived at below, be assumed to be zero or absorbed in the summation in Eq. (1) through the use of two additional always-on neurons that communicate to every other neuron in the net its threshold and external input levels, through appropriate weights added to T_{ij}. Note in Fig. 1 that the iterated input vector is always the transpose of the thresholded output vector.

By substituting the expression for the storage matrix

$$ T_{ij} = \sum_{m=1}^{M} v_i^{(m)} v_j^{(m)}, \tag{2} $$

formed by summing the outer products of the stored vectors $v_i^{(m)}$, $i = 1,2, \ldots N$ and $m = 1,2 \ldots M$, into Eq. (1) and interchanging the order of summations, we obtain

$$ {}^{(q+1)}v_i = \operatorname{sgn}\left\{\sum_{m=1}^{M} {}^{(q)}C_m v_i^{(m)}\right\}, \tag{3} $$

where

$$ {}^{(q)}C_m = \sum_{j=1}^{N} v_j^{(m)}\,{}^{(q)}v_j \tag{4} $$

are coefficients determined by the inner product of the input vector ${}^{(q)}\bar{v}$ at any iteration by each of the stored vectors. Equations (3) and (4) can be implemented employing the optoelectronic direct storage and inner-product recall scheme shown in Fig. 2 in which LEA and PDA represent light emitting array and photodetector array, respectively. Noting that the two segments to the left and to the right of the diffuser in Fig. 2 are identical, one can arrive at the simplified equivalent reflexive inner-product scheme shown in Fig. 3. Now we have arrived at two equivalent implementations of the neural model. These are shown together in Fig. 4. One employs outer-product distributed storage and vector–matrix multiplication with thresholded feedback in the recall as shown in Fig. 4(a), and the second employs direct storage and inner-product recall with thresholded feedback as shown in Fig. 4(b). The reflexive or inner-product scheme has several advantages over the outer-product scheme. One is storage capacity. While an $N \times N$ storage matrix in the outer-product scheme can store $M \leq N/4 \ln N$ vectors of length N beyond which the probability of correct recall deteriorates rapidly because of proliferation of spurious states,[32] the storage mask T_{jm}, j 1,2 ... N, m

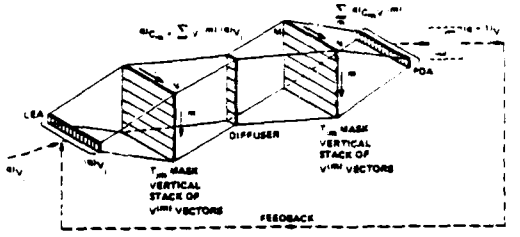


Fig. 2. Direct storage and inner-product recall scheme.

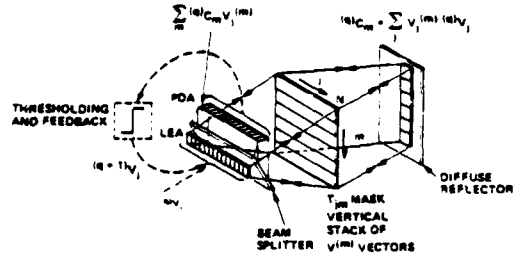


Fig. 3. Reflexive inner-product scheme.

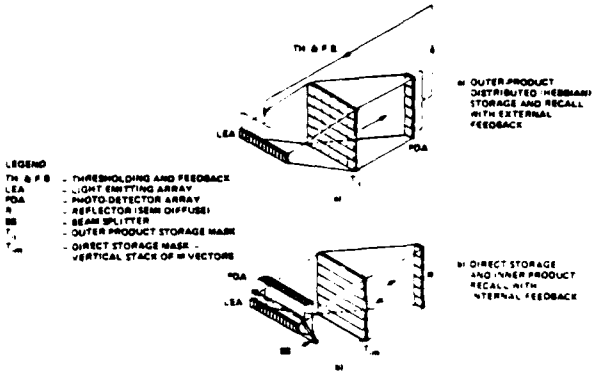


Fig. 4. Two equivalent neural net analogs: (a) outer-product distributed storage and recall with external feedback; (b) reflexive inner-product direct storage and recall with internal feedback.

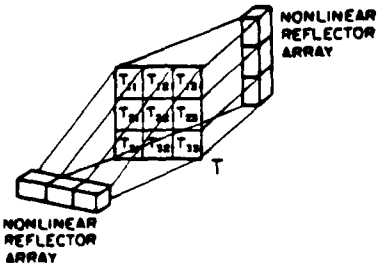


Fig. 5. Concept of nonlinear resonator content addressable memory.

$= 1,2 \ldots M$, in the inner-product scheme can store directly a stack of up to $\mathbf{M} = \mathbf{N}$ vectors in the same size matrix. It can be argued that the robustness and the fault tolerance of the distributed storage scheme have been sacrificed in the inner-product scheme, but this is a mute argument. Robustness can be easily restored by introducing a certain degree of redundancy in the inner-product scheme of Fig. 4(b). This can be done, for example, by storing a vector in more than one location in the stack. The internal optical feedback in the inner-product implementation is certainly another attractive advantage. In fact the beauty of internal feedback has inspired the concept of reflexive associative memory or nonlinear resonator CAM (content addressable memory) shown in Fig. 5. This scheme, which becomes possible because the T matrix is symmetrical, utilizes the same optics for internal feedback and for transposing the reflected state vectors. The scheme is perfectly suited for use with nonlinear reflector arrays or arrays of optically bistable elements. The advantages of a similar bidirectional associative memory have also been noted recently elsewhere.[25]

In view of the obvious advantages of the reflexive scheme [Fig. 4(b)], one is led to question the reason nature appears to prefer distributed (Hebbian) storage [as in Fig. 4(a)] over localized storage [as in Fig. 4(b)] besides fault tolerance and redundancy. As a result of the preceding exercise the answer now comes readily to mind: in the inner-product scheme the connectivity matrix T_{ij} is not present. Self-organization and learning in biological systems are associated with modifications of the synaptic weights matrix. Hence learning in the neural sense is not possible in the inner-product scheme. In this sense the inner-product scheme of Fig. 4(b) is not neural but involves conventional correlations between the input vectors and the stored vectors. One can argue that the instant the identity of the weights matrix T_{ij} was obliterated the inner product network stopped being neural as learning through weights modification is no longer possible. We are therefore led to conclude that distributed storage and self-organization and learning are the most distinctive features of neural signal processing as opposed to conventional approaches to signal processing such as in the inner-product scheme which involves simple correlations and where it is not clear how self-organization and learning can be performed since there is no T_{ij} matrix to be modified.

Neural net processing has additional attractive features that are not as distinctive as self-organization and learning. These include heteroassociative storage and recall where the same net performs the functions of storage, processing, and labeling of the output (final state) simultaneously. While such a task may also be realized with conventional signal processing nets, each of the above three functions must however be realized separately in a different subnet. A striking example of this feature reported recently[23] is in the area of radar target recognition from partial information employing sinogram representation of targets of interest. The sinogram representations were used in computing and

setting the synaptic weight matrix in an explicit learning mode. Recognition in radar from partial information is tantamount to solution of the superresolution problems. The ease and elegance with which the neural net approach solves this classical problem is, to say the least, impressive.

Other distinctive features of neural nets associated with the rich phase-space behavior are bifurcation and chaotic behavior. These were mentioned earlier but are restated here because of their importance in sequential processing of data (e.g., cyclic heteroassociative memory) and in the modeling and study of mental disorder and the effect of drugs on the nervous system.[33]

III. Partitioning Architectures and Stochastic Learning by Simulated Annealing

In preceding work on optical analogs of neural nets,[10–25] the nets described were programmed to do a specific computational task, namely, a nearest-neighbor search that consisted of finding the stored entity that is closest to the address in the Hamming sense. As such the net acted as a content addressable associative memory. The programming was done by first computing the interconnectivity matrix using a Hebbian (outer-product) recipe given the entities one wished the net to store, followed by setting the weights of synaptic interconnections between neurons accordingly.

In this section we are concerned with architectures for optoelectronic implementation of neural nets that are able to program or organize themselves under supervised conditions. Such nets are capable of (a) computing the interconnectivity matrix for the associations they are to learn, and (b) changing the weights of the links between their neurons accordingly. Such self-organizing networks therefore have the ability to form and store their own internal representations of the associations they are presented with. The discussion in this section is an expansion of one given earlier.[34]

Multilayered self-programming nets have recently been attracting increasing attention.[4,28–30,35] For example, in Ref. 28 the net is partitioned into three groups, two are input and output groups of neurons that interface with the net environment and the third is a group of hidden or internal units that acts as a buffer between the input and output units and participates in the process of forming internal representations of the associations the net is presented with. This can be done, for example, by clamping or fixing the states of neurons in the input and output groups to the desired pairs of associations and letting the net run through its learning algorithm to arrive ultimately at a specific set of synaptic weights or links between the neurons. No neuron or unit in the input group is linked directly to a neuron in the output group and vice versa. Any such communication must be carried out via the hidden units. Neurons within the input group can communicate among each other and with hidden units and the same is true for neurons in the output group. Neurons in the hidden group cannot communicate among each other. They can only communicate with neurons in the input and output groups as stated earlier.

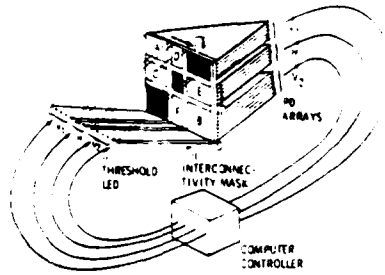


Fig. 6. Optoelectronic analog of self-organizing neural net partitioned into three layers capable of stochastic self-programming and learning.

Two supervised learning procedures in multilayered nets have recently attracted attention. One is stochastic, involving a simulated annealing process,[26,27] and the other is deterministic, involving an error back-propagation process.[30] There is general agreement, however, that because of their iterative nature, sequential computation of the weights using these algorithms is very time-consuming. A faster means for carrying out the required computations is needed. Nevertheless, the work mentioned represents a milestone in that it opens the way for powerful collective computations in multilayered neural nets and the partitioning concept dispels earlier reservations[36] about the capabilities of early single layered models of neural nets such as the Perceptron.[37] The partitioning feature and the ability to define input and output neurons may also be the key for realizing meaningful interconnection between neural modules for the purpose of performing higher-order hierarchical processing.

Optics and optoelectronic architectures and techniques can play an important role in the study and implementation of self-programming networks and in speeding up the execution of learning algorithms. Here we describe a method for partitioning an optoelectronic analog of a neural net to implement a multilayered net that can learn stochastically by means of a simulated annealing learning algorithm in the context of a Boltzmann machine formalism (see Fig. 6). The arrangement shown in Fig. 6 derives from the neural network analogs we described earlier.[12] The network, consisting of, say, N neurons, is partitioned into three groups. Two groups, V_1 and V_2, represent visible units that can be viewed as input and output groups, respectively. The third group H are hidden or internal units. The partition is such that $N_1 + N_2 + N_3 = N$, where N_1, N_2, and N_3 refer to the number of neurons in the V_1, V_2, and H groups, respectively. The interconnectivity matrix, T_{ij}, is partitioned into six submatrices, A, B, C, D, E, F, and three zero-valued submatrices shown as blackened or opaque regions of the T_{ij} mask. The LED array represents the state of the neurons, assumed to be unipolar binary (LED on = neuron firing,

LED off = neuron not firing). The T_{ij} mask represents the strengths of interconnection between neurons in a manner similar to earlier arrangements.[12] Light from each LED is smeared vertically over the corresponding column of the T_{ij} mask with the aid of an anamorphic lens system (not shown in Fig. 6), and light emerging from each row of the mask is focused with the aid of another anamorphic lens system (also not shown) onto the corresponding elements of the photodetector (PD) array. The same scheme utilized in Ref. 12 for realizing bipolar values of T_{ij} in incoherent light is assumed here, namely, separating each row of the T_{ij} mask into two subrows and assigning positive-valued T_{ij} to one subrow and negative-valued T_{ij} to the other, and focusing light emerging from the two subrows separately on two adjacent photosites on the photodetector array connected in opposition. Submatrix A, with $N_1 \times N_1$ elements, provides the interconnection weights between units or neurons within group V_1. Submatrix B, with $N_2 \times N_2$ elements, provides the interconnection weights between units within V_2. Submatrices C (with $N_1 \times N_3$ elements) and D (with $N_3 \times N_1$ elements) provide the interconnection weights between units of V_1 and H and submatrices E (with $N_2 \times N_3$ elements) and F (with $N_3 \times N_2$ elements) provide the interconnection weights of units of V_2 and H. Units in V_1 and V_2 cannot communicate among each other directly because locations of their interconnectivity weights in the T_{ij} matrix or mask are blocked out (blackened lower left and top right portions of T_{ij}). Similarly units within H do not communicate among each other because locations of their interconnectivity weights in the T_{ij} mask are also blocked out (blackened center square of T_{ij}). The LED element θ is of graded response. Its output represents the state of an auxiliary neuron in the net that is always on to provide a global threshold level to all units by contributing only to the light focused onto negative photosites of the photodetector (PD) arrays from pixels in the G column of the interconnectivity mask. This is achieved by suitable modulation of the transmittance of pixels in the G column. This method for introducing the threshold level is attractive, as it allows for providing to all neurons in the net a fixed global threshold, an adaptive global threshold, or even nosiy global threshold if desired.

By using a computer-controlled nonvolatile spatial light modulator to implement the T_{ij} mask in Fig. 6 and including a computer controller as shown, the scheme can be made self-programming with ability to modify the weights of synaptic links between its neurons. This is done by fixing or clamping the states of the V_1 (input) and V_2 (output) groups to each of the associations we want the net to learn and by repeated application of the simulated annealing procedure with Boltzmann, or other stochastic state update rule, and collection of statistics on the states of the neurons at the end of each run when the net reaches thermodynamic equilibrium.

Stochastic learning by simulated annealing in the partitioned net proceeds as follows:

(1) Starting from an arbitrary T_{ij} clamp V_1 and V_2 to the desired association keeping H free running.

(2) Randomly select a neuron in H, say the kth neuron, and flip its state [recall we are dealing with binary (0,1) neurons].

(3) Determine the change ΔE_k in global energy E of the net caused by changing the state of the kth neuron.

(4) If $\Delta E_k < 0$, adopt the change.

(5) If $\Delta E_k > 0$, do not discard the change outright but calculate first the Boltzmann probability factor,

$$P_k = \exp \frac{-\Delta E_k}{T}. \tag{5}$$

and compare the outcome to a random number $N_r \in [0,1]$. If $P_k > N_r$, adopt the change of states of the kth neuron even if it leads to an energy increase (i.e., $\Delta E_k > 0$). If $P_k < N_r$, discard change, i.e., return the kth neuron to its original state.

(6) Once more select a neuron in H randomly and repeat steps (1)–(5).

(7) Repeat steps (1)–(6) reducing at every round the temperature T gradually [e.g., $T = T_0/\log(1 + m)$, where m is the round number, cooling schedule is frequently used to ensure convergence] until a situation is reached where changing states of neuron in H does not alter the energy E, i.e., $\Delta E_k \to 0$. This indicates a state of thermodynamic equilibrium or a state of global energy minimum has been reached. The temperature T determines the fineness of search for a global minimum. A high T produces coarse search and low T a finer grained search.

(8) Record the state vector at thermodynamic equilibrium, i.e., the states of all neurons in the net, i.e., those in H and those in V_1 and V_2 that are clamped.

(9) Repeat steps (1)–(8) for all other association on V_1 and V_2 we want the net to learn and collect statistics on the states of all neurons by storing the states at thermodynamic equilibrium in computer memory as in step (8). This completes the first phase of exposing the net to its environment.

(10) Generate the probabilities P_{ij} of finding the ith neuron and the jth neuron in the same state. This completes phase I of the learning cycle.

(11) Unclamp neurons in V_2 letting them run free as with neurons in H.

(12) Repeat steps (1)–(10) for all input vectors V_i and collect statistics on the states of all neurons in the net.

(13) Generate the probabilities P'_{ij} of finding neuron i and neuron j in the same state.

(14) Increment the current connectivity matrix T_{ij} by $\Delta T_{ij} = \epsilon(P_{ij} - P'_{ij})$ where ϵ is a constant representing and controlling the speed of learning. This completes phase II of the learning cycle.

(15) Repeat steps (1)–(14) again and again until the increments ΔT_{ij} tend to zero, i.e., become smaller than some prescribed small number. At this point the net is said to have captured the underlying structure or formed its own representations of its environment defined by the associations presented to it. We are now dealing with a learned net.

One can make the following observations regarding the above procedure:

The search for state of global energy minimum is basically a gradient descent procedure that allows for probabilistic hill climbing to avoid entrapment in a state of local energy minimum. The relative probability of two global states α and β is given by the Boltzmann distribution $P_\alpha/P_\beta = \exp\{-(E_\alpha - E_\beta)/T\}$, hence the name Boltzmann machine.[28] Therefore the lowest energy state is the most probable at any temperature and is sought by the procedure.

Unlike explicit programming of a neural net where lack of correlation among the stored vectors is a prerequisite for ideal storage and recall, self-programming by simulated annealing has no such requirement. In fact learning by simulated annealing in a Boltzmann machine looks for underlying similarities or correlations in the training set to generate weights that can make the net generalize. Generalization is a property where the net recognizes an entity presented to it even though it was not among those specifically used in the learning session. Learning is thus not rote.

The final T_{ij} reached represents a net that has learned its environment by itself under supervision, i.e., it has formed its own internal representations of its surroundings. Those environmental states or input/output associations that occur more frequently will influence the final T_{ij} more than others and hence form more vivid impressions in the synaptic memory matrix T_{ij}.

The learning procedure is stochastic but is still basically Hebbian in nature where the change in the synaptic interconnection between two units (neurons) depends on finding the two units in the same state (sameness reinforcement rule).

Evidently, being stochastic in nature (involving probabilistic state transition rules and simulated annealing) the learning procedure is lengthy (taking hours in a digital simulation for nets of a few tens to a few hundred neurons). Hence, speeding up the process by using analog optoelectronic implementation is highly desirable.

Stochastic learning consists of two phases: phase I involves generating probabilities P_{ij} when the input and output of the net are specified. Phase II involves generating the probabilities P'_{ij} when only the input is specified while the rest of the net is free running followed by computing the weight increments and modifying the T_{ij} matrix accordingly.

IV. Accelerated Learning

Stochastic learning by the simulated annealing procedure we described was originally conceived for serial computation. When dealing with parallel optical computing systems it does not make sense to exactly follow a serial algorithm. Modifications that can take advantage of the available parallelism of optics to speed up stochastic learning are therefore of interest. In this section we discuss several such modifications that offer potential for speeding-up stochastic learning in optoelectronic implementations by several orders of magnitude compared to serial digital implementation.

Learning by simulated annealing requires calculating the energy E of the net,[7,38]

$$E = \frac{1}{2}\sum_i u_i v_i, \tag{6}$$

where v_i is the state of the ith neuron and

$$u_i = \sum_{j \neq i} T_{ij} v_j - \theta_i + I_i \tag{7}$$

is the activation potential of the ith neuron with θ_i and I_i being the threshold level and external input to the i-th neuron respectively and the summation term representing the input to the i-th neuron from all other neurons in the net. By absorbing θ_i and I_i in the summation term as described earlier, Eq. (7) can be simplified to

$$u_i = \sum_{j \neq i} T_{ij} v_j. \tag{8}$$

A simple analog circuit for calculating the contribution E_i of the ith neuron to the global energy E of the net is shown in Fig. 7(a). Here the product of the activation potential of the ith neuron and the state v_i of the ith neuron is formed to obtain E_i which is then added to all terms formed similarly in parallel for all other neurons in the net. Although VLSI implementation of such an analog circuit for parallel calculation of the global energy is feasible, this becomes less attractive as the number of neurons increases because of the interconnection problem associated with the large fan-in at the summation element.

A simplified version of a rapid scheme for obtaining E optoelectronically is shown in Fig. 7. The scheme requires the use of an electronically addressed nonvolatile binary (on–off) spatial light modulator consisting of a single column of N pixels. A suitable candidate is a parallel addressed magnetooptic spatial light modu-

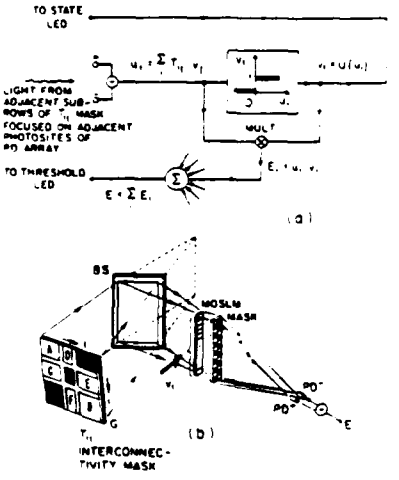


Fig. 7. Two schemes for parallel computing of the global energy in an optoelectronic analog of a multilayered self-organizing net: (a) electronic scheme; (b) optoelectronic scheme.

lator (MOSLM) consisting of a single column of N pixels that are driven electronically by the same signal driving the LED array to represent the state vector $\bar{\mathbf{v}}$ of the net. A fraction of the focused light emerging from each row of the T_{ij} mask is deflected by the beam splitter (BS) onto the individual pixels of the column MOSLM such that light from adjacent pairs of subrows of T_{ij} falls on one pixel of the MOSLM. The MOSLM pixels are overlaid by a checkered binary mask as shown. The opaque and transparent pixels in the checkered mask are staggered in such a fashion that light emerging from the left subcolumn will originate from the positive subrows T_{ij}^{+} of T_{ij} only and light emerging from the right subcolumn will originate from the negative subrows T_{ij}^{-} or T_{ij}. By separately focusing the light from the left and right subcolumns as shown onto two photodetectors and subtracting and halving their outputs, one obtains

$$E = -\frac{1}{2}\sum_i\left[\left(\sum_{j\neq i} T_{ij}^{+} - T_{ij}^{-}\right)\mathbf{v}_j\right]\mathbf{v}_i$$

$$= -\frac{1}{2}\sum_i\left(\sum_{j\neq i} T_{ij}\mathbf{v}_j\right)\mathbf{v}_i = -\frac{1}{2}\sum_i \mathbf{u}_i\mathbf{v}_i, \qquad (9)$$

which is the required global energy.

The learning procedure detailed in Sec. III requires fast random number generation for use in random drawing and switching of state of neurons from H (during phase I of learning) and from H and V_2 (during phase II of learning). Another random number is also needed to execute the stochastic state update rule when $\Delta E_k > 0$. Although fast digital pseudorandom number generation of up to $10^9\ s^{-1}$ is feasible[39] and can be used to help speed up digital simulation of the learning algorithm, this by itself is not sufficient to make a large impact especially when the total number of neurons in the net is large. Optoelectronic random number generation is also possible although at a slower rate of 10^5 s. Despite the slower rate of generation, optoelectronic methods have advantages that will be elaborated on below. An optoelectronic method for generating the Boltzmann probability factor $p(\Delta E_k)$ [see Eq. (5)] employing speckle statistics is described in Ref. 40 and optical generation of random number arrays by photon counting image acquisition systems or clipped laser speckle have also been recently described.[41–44] These photon counting image acquisition systems have the advantage of being able to generate normalized random numbers with any probability density function. A more important advantage of optical generation of random number arrays however is the ability to exploit the parallelism of optics to modify the simulated annealing and the Boltzmann machine formalism detailed above to achieve significant improvement in speed. As stated earlier, with parallel optical random number generation, a spatially and temporally uncorrelated linear array of perculating light spots of suitable size can be generated and imaged on the photodetector array (PDA) of Fig. 6 such that both the positive and negative photosites of the PDA [see also Fig. 7(a)] are subjected to random irradiance. This introduces a random (noise) component in θ_i and I_i of Eq. (7) which can be viewed as a bipolar noisy threshold. The noisy threshold produces in turn a noisy component in the energy in accordance with Eq. (6). The magnitude of the noise components can be controlled by varying the standard deviation of the random light intensity array irradiating the PDS. The noisy threshold therefore produces random controlled perturbation or shaking of the energy landscape of the net. This helps shake the net loose whenever it gets trapped in a local energy minimum. The procedure can be viewed as generating a controlled deformation or tremor in the energy landscape of the net to prevent entrapment in a local energy minimum and thereby ensure convergence to a state of global energy minimum. Both the random drawing of neurons (more than one at a time is now possible) and the stochastic state update of the net are now done in parallel at the same time. This leads to significant acceleration of the simulated annealing process. The parallel optoelectronic scheme for computing the global energy described earlier [see Fig. 7(b)] can be used to modulate the standard deviation of the optical random noise array used to produce a noisy threshold with a

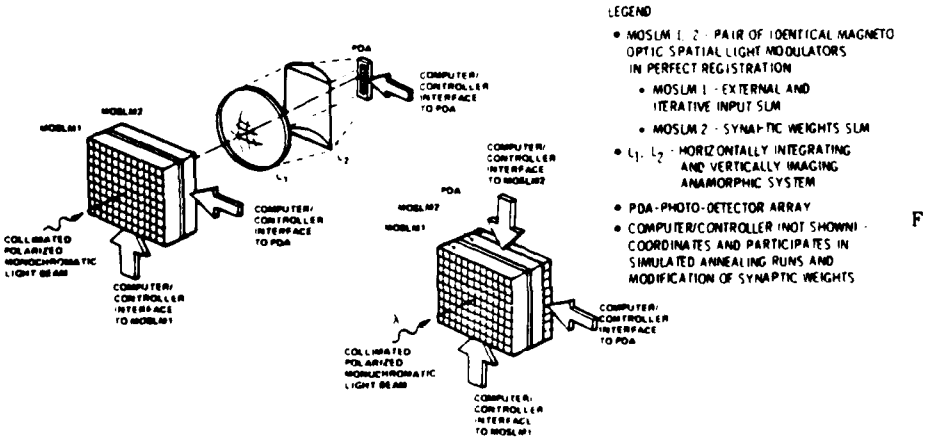


Fig. 8. Optoelectronic neural chip.

function of the instantaneous global energy E and/or its time rate change dE/dt. In this fashion an adaptive noisy threshold scheme can be realized to control the tremors in the energy landscape if necessary. The above discussion gives an appreciation of the advantages and flexibility of using optical random array generators in making the net rapidly find states of global energy minimum. No attempt is made here to estimate in detail the speed enhancement over digital execution of the simulated annealing process as this will be dependent on the characteristics of the light emitting array, the photodetector array, the spatial light modulator, and the speed of the computer–controller interface used. Nevertheless, the enhancement over digital serial computation can be significant, approaching 5–6 orders of magnitude especially for relatively large multilayer nets consisting of from a few tens to a few hundred neurons. A recent study of learning in neuromorphic VLSI systems in the context of a modified Boltzmann machine gives speedup estimates of 10^6 over serial digital simulations.[45]

V. Optoelectronic Neural Chip

The discussion in the preceding sections shows that optical techniques can simplify and speedup stochastic learning in artificial neural nets and make them more practical. The attractiveness and practicality of optoelectronic analogs of self-programming and learning neural nets are enhanced further by the concept of optoelectronic neural chips presented in Fig. 8. The embodiments shown rely heavily on the use of computer or microprocessor interfaced spatial light modulators and photodetector arrays. The figure shows how the free-space anamorphic lens system in the top left embodiment can be replaced by a single photodetector array with horizontal strip elements that spatially integrate the light emerging from rows of MOSLM 2 (lower right embodiment). MOSLM 2 represents the T_{ij} mask of Fig. 6. Each column MOSLM 1 is uniformly activated by the computer controller. This replaces the function of the anamorphic lens system that was needed in Fig. 6 to smear the light from the LED array vertically onto the elements of the T_{ij} mask. The optoelectronic neural chip represents a neural module operating in an ambient light environment as compared with a biological neural module operating in a chemical environment. The chip thus derives some of its operating energy from the ambient light environment.

VI. Discussion

The architecture described here for partitioning a neural net can be used in hardware implementation and study of self-programming and learning algorithms such as, for example, the simulated annealing algorithm outlined here. The parallelism and massive interconnectivity provided through the use of optics should markedly speed up learning even for the simulated annealing algorithm, which is known to be quite time-consuming when carried out on a sequential machine. The partitioning concept described is also extendable to multilayered nets of more than three layers and to 2-D arrangement of synaptic inputs to neurons, as opposed to the 1-D or lineal arrangement described here. Other learning algorithms calling for a multilayered architecture such as the error backprojection algorithm[30] and its coherent optics implementation[45] can also now be envisioned optoelectronically employing the partitioning scheme described here.

Learning algorithms in layered nets lead to analog or multivalued T_{ij}. Therefore high-speed computer-controlled SLMs with graded pixel response are called for. Methods of reducing the needed dynamic range of T_{ij} or for allowing the use of ternary T_{ij} are however under study to enable the use of commercially available fast nonvolatile binary SLM devices such as the Litton/Semetex magnetooptic SLM (MOSLM).[46] A frame switching time better than 1/1000 s has been demonstrated recently in our work on a 48×48 pixel device by employing an external magnetic field bias. It is worth noting that the role of optics in the architecture described not only facilitates partitioning the net into groups or layers but also provides the massive interconnectivity mentioned earlier. For example, for a neural net with a total of $N = 512$ neurons, the optics enable making $2N^2 = 2.62 \times 10^5$ programmable weighted interconnections among the neurons in addition to the $4N = 2048$ interconnections that would be needed in Fig. 6(b) to compute the energy E.

Assuming that material and device requirements of the architectures described can be met and partitioned self-organizing neural net modules will be routinely constructed, the addition of such a module to a computer controller through a high speed interface can be viewed as providing the computer controller with artificial intelligence capabilities by imparting to it neural net attributes. These capabilities include self-organization, self-programmability and learning, and associative memory capability for conducting nearest-neighbor searches. Such attributes would enable a small computer to perform powerful computational tasks of the kind needed in pattern recognition, and in the solution of combinatorial optimization problems and ill-posed problems encountered, for example, in inverse scattering and vision, which are confined at present to the domain of supercomputers.

A central issue in serial digital computation of complex problems is computational complexity.[47] Programming a serial computer to perform a complex computational task is relatively easy. The computation time however for certain problems, especially those dealing with combinatorial searches and combinatorial optimization, can be extensive. In neural nets the opposite is true. They take time to program [for example, computation of the interconnectivity matrix of synaptic weights by outer product or correlation (Hebbian rule) and setting the weights accordingly]. Once programmed, however, they perform the computations required almost instantaneously. This fact is one of the first attributes noted when working with neural nets and has recently been elaborated on.[48]

Self-organization and learning entails the net deter-

mining by itself the weights of synaptic interconnections among its neurons that represent the association it is supposed to learn. In other words, the net programs itself, thereby alleviating the programming complexity issue. One can envision nets that learn by example when the associations the net is supposed to learn are presented to it by an external teacher in a supervised learning mode. This leads naturally to the more intriguing question of unsupervised learning in such nets and analog implementations of such learning.

This work was supported by grants from DARPA/NRL, the Army Research Office, and the University of Pennsylvania Laboratory for Research on the Structure of Matter. Some of the work reported was carried out when the author was a summer Distinguished Visiting Scientist at the NASA Jet Propulsion Laboratory in Pasadena. The support and hospitality of JPL are gratefully acknowledged.

References

1. B. Widrow, and M. E. Hoff, "Adaptive Switching Circuits," in WESCON Convention Board, Part 4 (1960), pp. 96–104; in *Self-Organizing Systems*, M. C. Yovitz *et al*, Eds. (Spartan, Washington, DC, 1962).
2. K. Nakano, "Associatron—a Model of Associative Memory," IEEE Trans. Syst. Man Cybern. SMC-2, 380 (1972).
3. T. Kohonen, "Correlation Matrix Memories," IEEE Trans. Comput. C-21, 353 (1972).
4. T. Kohonen, *Associative Memory* (Springer-Verlag, Heidelberg, 1978); *Self-Organization and Associative Memory* (Springer-Verlag, New York, 1984).
5. D. J. Willshaw, "A Simple Network Capable of Inductive Generalization," Proc. R. Soc. London 182, 233 (1972).
6. J. A. Anderson, J. W. Silverstein, S. A. Ritz, and R. S. Jones, "Distinctive Features, Categorial Perception, and Probability Learning: Some Applications of a Neural Model," Physiol. Rev. 34, 413 (1977).
7. J. J. Hopfield, "Neural Networks and Physical Systems with Emergent Collective Computational Abilities," Proc. Natl. Acad. Sci. U.S.A. 79, 2554 (1982); "Neurons with Graded Response Have Collective Computational Properties Like Those of Two-State Neurons," Proc. Natl. Acad. Sci. U.S.A. 81, 3088, (1984).
8. S. Grossberg, *Studies of Mind and Brain* (Reidel, Boston, 1982).
9. A. C. Sanderson and Y. Y. Zeevi, Eds., Special Issue on Neural and Sensory Information Processing," IEEE Trans. Syst. Man Cybern. SMC-13 (1983).
10. D. Psaltis and N. Farhat, "A New Approach to Optical Information Processing Based on the Hopfield Model," in *Digest of the Thirteenth Congress of the International Commission on Optics* (ICO-13), Sapporo, Japan (1984), p. 24.
11. D. Psaltis and N. Farhat, "Optical Information Processing Based on an Associative-Memory Model of Neural Nets with Thresholding and Feedback," Opt. Lett. 10, 98 (1985).
12. N. H. Farhat, D. Psaltis, A. Prata, and E. Paek, "Optical Implementation of the Hopfield Model," Appl. Opt. 24, 1469 (1985).
13. N. H. Farhat and D. Psaltis, "Architectures for Optical Implementation of 2-D Content Addressable Memories," in *Technical Digest, Optical Society of America Annual Meeting* (Optical Society of America, Washington, DC, 1985), paper WT3.
14. K. S. Lee and N. H. Farhat, "Content Addressable Memory with Smooth Transition and Adaptive Thresholding," in *Technical Digest, Optical Society of America Annual Meeting* (Optical Society of America, Washington, DC, 1985), paper WJ35.
15. D. Psaltis, E. Paek, and J. Hong, "Acoustooptic Implementation of Neural Network Models, in *Technical Digest, Optical Society of America Annual Meeting* (Optical Society of America, Washington, DC, 1985), paper WT6.
16. J. D. Condon, "Optical Window Addressable Memory," in *Technical Digest of Topical Meeting on Optical Computing* (Optical Society of America, Washington, DC, 1985), postdeadline paper.
17. A. D. Fisher, C. L. Giles, and J. N. Lee, "Associative Processor Architectures for Optical Computing," J. Opt. Soc. Am. A 1, 1337 (1984); "An Adaptive, Associative Optical Computing Element," in *Technical Digest of Topical Meeting on Optical Computing* (Optical Society of America, Washington, DC, 1985), paper WB4.
18. A. D. Fisher and C. L. Giles, "Optical Adaptive Associative Computer Architectures," in *Proceedings, IEEE COPCOM Spring Meeting*, IEEE catalog no. CH2135-2/85/0000/0342 (1985), p. 342.
19. A. D. Fisher, "Implementation of Adaptive Associative Optical Computing Elements," Proc. Soc. Photo-Opt. Instrum. Eng. 625, 196 (1986).
20. B. H. Soffer, G. J. Dunning, Y. Owechko, and E. Marom, "Associative Holographic Memory with Feedback Using Phase-Conjugate Mirrors," Opt. Lett. 11, 118 (1986).
21. D. Z. Anderson, "Coherent Optical Eigenstate Memory," Opt. Lett. 11, 56 (1986).
22. A. Yariv and S-K. Kwong, "Associative Memories Based on Message-Bearing Optical Modes in Phase-Conjugate Resonators," Opt. Lett. 11, 186 (1986).
23. N. Farhat, S. Miyahara, and K. S. Lee, "Optical Implementation of 2-D Neural Nets and Their Application in Recognition of Radar Targets," in *Neural Networks for Computing*, J. S. Denker, Ed. (American Institute of Physics, New York, 1986), p. 146.
24. N. Farhat and D. Psaltis, "Optical Implementation of Associative Memory Based on Models of Neural Networks," in *Optical Signal Processing*, J. L. Horner, Ed. (Academic, New York, 1987), p. 129, in press.
25. B. Kosko and C. Guest, "Optical Bidirectional Associative Memory," Proc. Soc. Photo-Opt. Instrum. Eng. 758, (1987), in press.
26. N. Metropolis, A. M. Rosenbluth, M. N. Rosenbluth, and A. H. Teller, "Equations of State Calculations by Fast Computing Machines," J. Chem. Phys. 21, 1087 (1953).
27. S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by Simulated Annealing," Science 220, 671 (1983).
28. D. H. Ackley, G. E. Hinton, and T. J. Seinowski, "A Learning Algorithm for Boltzmann Machines," Cognitive Sci. 1, 147 (1985).
29. T. J. Sejnowski and C. R. Rosenberg, "NETtalk: A Parallel Network That Learns to Read Loud," Johns Hopkins U., Electrical Engineering & Computer Science Technical Report JHU EECS-96/01 (1986).
30. D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning Internal Representations by Error Propagation," in *Parallel Distributed Processing*, D. F. Rumelhart and J. L. McClelland, Eds. (MIT Press, Cambridge, MA, 1986), Vol. 1, p. 318.
31. S. Miyahara, "Automated Radar Target Recognition Based on Models of Neural Networks," U. Pennsylvania Dissertation (1987).
32. R. J. McEliece, E. C. Posner, E. R. Rodemich, and S. S. Venkatesh, "The Capacity of the Hopfield Associative Memory," IEEE Trans. Inf. Theory IT-33, 461, March 1987.
33. G. B. Ermentrout and J. D. Cowan, "Large Scale Spatially Organized Activity in Neural Nets," SIAM (Soc. Ind. Appl. Math.) J. Appl. Math. 38, 1 (1980).
34. N. H. Farhat, "Architectures for Opto-Electronic Analogs of Self-Organizing Neural Networks," Opt. Lett. 12, 448 (1987).

accepted for publication.
35. G. A. Carpenter and S. Grossberg, "A Massively Parallel Architecture for Self-Organizing Neural Pattern Recognition Machines," Comput. Vision Graphics Image Process. **37**, 54 (1987).
36. M. L. Minsky and S. Papert, *Perceptrons* (MIT Press, Cambridge, MA, 1969).
37. F. Rosenblatt, *Principles of Neuro-Dynamics: Perceptions and the Theory of Brain Mechanisms* (Spartan Books, Washington, DC, 1962).
38. M. A. Cohen and S. Grossberg, "Absolute Stability of Global Pattern Formation and Parallel Memory Storage by Competitive Neural Networks," IEEE Trans. Syst. Man Cybern. **SMC-13**, 815 (1983).
39. S. Kirkpatrick, IBM Watson Research Center; private communication (1987).
40. A. J. Ticknor, H. H. Barrett, and R. L. Easton, Jr., "Optical Boltzmann Machines," in *Technical Digest of Topical Meeting on Optical Computing* (Optical Society of America, Washington, DC, 1985), postdeadline paper PD3.
41. G. M. Morris, "Optical Computing by Monte Carlo Methods," Opt. Eng. **24**, 86 (1985).
42. J. Marron, A. J. Martino and G. M. Morris, "Generation of Random Arrays Using Clipped Laser Speckle," Appl. Opt. **25**, 26 (1986).
43. F. Devos, P. Garda, and P. Chavel, "Optical Generation of Random-Number Arrays for On-Chip Massively Parallel Monte Carlo Cellular Processors," Opt. Lett. **12**, 152 (1987).
44. Y. Tsuchiya, E. Inuzuka, T. Kurono, and M. Hosada, "Photon Counting Imaging and Its Application," *Advances in Electronics and Electron Physics*, B. L. Morgan, Ed. 64A, 21 (Academic Press, London, 1988).
45. J. Alspector and R. B. Allen, "A Neuromorphic VLSI Learning System," in *Advanced Research in VLSI*, Paul Losleben, Ed. (MIT Press, Cambridge, MA, 1987), to be published.
46. W. E. Ross, D. Psaltis, and R. H. Anderson, "Two-Dimensional Magneto-Optic Spatial Light Modulator for Signal Processing," Opt. Eng. **22**, 485 (1983).
47. R. M. Karp, "Combinatorics, Complexity and Randomness," Commun. ACM **29**, 98 (Feb. 1986).
48. M. Takeda and J. W. Goodman, "Neural Networks for Computation: Number Representations and Programming Complexity," Appl. Opt. **25**, 3033 (1986).
49. D. G. Bounds, "Numerical Simulations of Boltzmann Machines," in *Neural Networks For Computing*, J. S. Denker, Ed., Vol. 151, (American Institute of Physics, New York, 1986), p. 59.

Appendix IV

Phased-Array Antenna Pattern Synthesis by Simulated Annealing

N. H. Farhat
B. Bai

Reprinted from: PROCEEDINGS OF THE IEEE, Vol. PROC-75, No. 6, JUNE 1987.

ferent steering angles, but an optimal weight distribution can be computed for each steering angle. From the simulation result, it is seen that the optimum far-field pattern has similar features to the pattern given by the Dolph-Chebyshev distribution function. In the Dolph-Chebyshev pattern, all the sidelobes have the same level for a specified beamwidth. A numerical example in [1] shows an 8-element array (element separation $d = 0.5\lambda$) with 25.8(dB) sidelobe level and 40.8° beamwidth. The optimum pattern given by our simulation shows nearly equal level sidelobes which are minimized for the given beamwidth.

V. Discussion

Simulated annealing is a modification of the iterative improvement algorithm [4]. It is physically more meaningful and can be computed more systematically than the iterative improvement [4]. Physically, the simulated annealing process is analogous to the cooling of melt in crystal growth: careful annealing produces a defect-free crystal, rapid annealing produces a defective crystal or glass [3]. The probabilistic treatment with the probability function $P(\Delta E) = \exp(-\Delta E/KT)$ provides a way to accept the unfavorable changes and is easy to compute. From our simulation, it has been found that the simulated annealing algorithm seems always to give better performance than the iterative improvement algorithm.

Since simulated annealing is a modified iterative improvement process, it takes a relatively long time to do an optimization problem just as iterative improvement does in a computer calculation. The phased-array synthesis in our simulation runs for 1 h or so for an array of 41 elements on a MICRO PDP-11 computer. Finding an efficient scheme to reduce the excessive amounts of computer time for most optimization problems has always been of concern [5]-[7]. Otherwise, if enough computation power is available, iterative improvement can be run from random starts for many times to approach the optimum state. Fast optodigital computing schemes similar to those described in [8] may also be considered for phased-array synthesis by simulated annealing. It is understood that the far field is the Fourier transform of the array distribution function. An optical lens can be used for computing the Fourier transform as the distribution function is inputted to the front focal plane of the lens via, for example, an appropriate computer-driven spatial light modulator (SLM). The Fourier transform in the back focal plane can be recorded and fed to the computer/controller to make the simulated annealing decision. The outcome is fedback to the SLM to change the distribution function in the front focal plane. The hybrid optodigital scheme will do the Fourier transform instantly. In this fashion, the computation associated with the Fourier transform can be virtually eliminated assuming a high-speed SLM and computer interface are utilized. An optoelectronic Boltzmann machine for accelerating the selection rule has also been proposed earlier in [8]. This process can be repeated for each step in simulated annealing. Also, a Cauchy probability selection rule, instead of the Boltzmann selection rule, can be used to speed up the whole annealing process further [7].

References

[1] C. L. Dolph, "A current distribution for broadside arrays which optimizes the relationship between beam width and side-lobe level," *Proc. IRE*, vol. 34, pp. 335-348, June 1946.

[2] N. Metropolis *et al.*, "Equation of state calculations by fast computing machines," *J. Chem. Phys.*, vol. 21, pp. 1087-1092, June 1953.

[3] S. Kirkpatrick *et al.*, "Optimization by simulated annealing," *Science*, vol. 220, pp. 671-680, May 1983.

[4] W. E. Smith *et al.*, "Reconstruction of objects from coded images by simulated annealing," *Opt. Let.*, vol. 8, pp. 199-201, Apr. 1983.

[5] B. Dunham, "Design by natural selection," *Synthese*, vol. 15, pp. 254-258, 1963.

[6] S. Lin, "Heuristic programming as an aid to network design," *Network*, vol. 5, pp. 33-43, 1975.

[7] H. Szu, "Fast simulated annealing with Cauchy probability density," presented at the Neural Networks for Computing Conference, Snowbird, UT, Apr. 1986.

[8] H. Barrett *et al.*, "Optical Boltzmann machines," post-deadline paper, OSA Topical Meeting on Optical Computing, Incline Village, NV, 1985.

0018-9219/87/0600-0844$1.00

Appendix V

Architectures and Methodologies for Self-Organization and Stochastic Learning in Opto-Electronic Analogs of Neural Nets

N.H. Farhat, and Z.Y. Shae
University of Pennsylvania
200 South 33rd Street
Philadelphia, PA 19104-6390

Published in the Proceedings of the IEEE First Annual International Conference on Neural Networks, June 1987.

IEEE Catalog #87TH0191-7

ARCHITECTURES AND METHODOLOGIES FOR SELF-ORGANIZATION AND STOCHASTIC LEARNING IN OPTO-ELECTRONIC ANALOGS OF NEURAL NETS

N.H. Farhat, and Z.Y. Shae
University of Pennsylvania
200 South 33rd Street
Philadelphia, PA 19104-6390

Abstract

Self-organization and learning is a distinctive feature of neural nets and processors that sets them apart from conventional approaches to signal processing. It leads to self-programmability which alleviates the problem of programming complexity in artificial neural nets. We have devised architectures for partitioning an opto-electronic analog of a neural net into distinct layers with prescribed inter-connectivity pattern to enable stochastic learning by simulated annealing in the context of a Boltzmann machine. Stochastic learning is of interest because of its relevance to the role of noise in biological neural nets. It can shed light on the way nature has turned noise present in biological nets to work to its advantage. Practical considerations and methodologies for appreciably accelerating stochastic learning in such a multi-layered net are also described. These include the use of parallel optical computation of the energy of the net, the use of fast nonvolatile programmable spatial light modulators to realize fast "plasticity", optical generation of random number arrays, and a noisy thresholding scheme that makes stochastic learning more biologically plausible and does not require determining the energy of the net for the annealing schedule.

1. INTRODUCTION

Interest in neural network models (see for example, [1]-[9]) and their optical analogs (see for example [10]-[21]) stems from well recognized information processing capabilities of the brain and the fit between what optics can do and what even simplified models of neural nets can offer toward the development of new approaches to collective signal processing that are robust, fault tolerant and can be extremely fast.

As a result opto-electronic analogs and implementations of neural nets are attracting today considerable attention. The optics in these implementations provide the needed parallelism and massive interconnectivity and therefore a potential for realizing relatively large neural nets while the decision making elements are realized electronically heralding a possible ultimate marriage of VLSI and optics.

Architectures suitable for use in the implementation of opto-electronic neural nets of one-dimensional and two-dimensional arrangements of neural nets have been studied and described recently [11]-[14]. Two-dimensional

architectures for opto-electronic analogs of neural nets have been successfully used in the recognition of objects from partial information by either complementing the missing information or by automatically generating correct labels of the data (object feature spaces) the memory is presented with [22].

In associative memory applications, the strength of interconnection between the "neurons" of the net is determined by the entities one wishes to store in the net. Specific storage "recipes" based on a Hebbian model of learning (outer-product storage algorithm) or variations thereof are usually employed. In that sense the memory is taught what is should know and be cognizant of. What can be of great utility however, is that neural nets can also be made to be self-organizing and learning i.e., to become self-programming. The combination of neural nets, Boltzmann machines, and simulated annealing concepts with high speed opto-electronic implementations promise to produce high-speed artificial neural net processors with stochastic rather than deterministic rules for decision making and state update that can form their own internal representations (connectivity weights) of their environment, the outside world data they are presented with, in a manner very analogous to the way the brain forms its own symbolic representations of reality. This is quite intriguing and has far reaching implications for smart sensing and recognition, thinking machines, and artificial intelligence as a whole. Our exploratory work is showing that optics can also play a role in the implementation and speeding up of learning algorithm (such as simulated annealing in the context of a Boltzmann machine formalism [23]-[26] and error back propagation [27]) in such self-teaching nets and for their subsequent use in automated robust recognition of entities the nets have had a chance to learn earlier by repeated exposure to them when in a learning mode. Self-organization and learning seems to be what sets apart optical and opto-electronic architectures and processing based on models of neural nets from other conventional approaches to optical processing.

In this paper we are therefore first concerned with architectures for opto-electronic implementation of neural nets that are able to program or organize themselves under supervised conditions, i.e., of nets that are capable of (a) computing the interconnectivity matrix for the associations they are to learn, and (b) of changing the weights of the links between their neurons accordingly. Such self-organizing networks have therefore the ability to form and store their own internal representations of the entities
or associations they are presented with. We are also concerned with stochastic learning in such nets and with methodologies for accelerating the learning process. These include a novel noisy threshold scheme that can speed up the simulated annealing process in opto-electronic analogs of neural nets. Results of computer simulations demonstrating capabilities of annealing with the noisy thresholding are presented.

2. PARTITIONING ARCHITECTURES AND STOCHASTIC LEARNING

Multi-layered self-programming nets have been described recently, [25], [26], where the net is partitioned into three groups. Two are groups of visible or external input/output units or neurons that interface with the

net environment or surroundings. The third is a group of hidden or internal units that separates the input and output units and participates in the process of forming internal representations of the associations the net is presented with, as for example by "clamping" or fixing the states of the input and output neurons to the desired associations and letting the net run through its learning algorithm to arrive ultimately at a specific set of synaptic weights or links between the neurons that capture, after many iterations of the process, the underlying structure of all the associations presented to the net. The hidden units or neurons prevent the input and output units from communicating with each other directly. In other words no neuron or unit in the input group is linked directly to a neuron in the output group and vice-versa. Any such communication must be carried out via the hidden units. Neurons within the input group do not communicate with each other. They can only communicate with neurons in the input and output groups as stated earlier.

As an example of the continuing role for optics, we describe next a concept for partitioning an opto-electronic analog of a neural net into input, output, and internal units with the selective communication pattern described above in order to realize a multi-layered net analog capable of stochastic learning, by means of a **simulated annealing** learning algorithm in the context of a **Boltzmann machine** formalism (see Fig. 1(a)). The arrangement shown derives from the neural network analogs we described earlier [11]. The network, consisting of say N neurons, is partitioned into three groups. Two groups, V_1 and V_2, represent visible or exterior units that can be used as input and output units respectively. The third group H are hidden or internal units. The partition is such that $N_1+N_2+N_3=N$ where subscripts 1, 2, 3 on N refer to the number of neurons in the V_1, V_2 and H groups respectively. The interconnectivity matrix, designated here as W_{ij}, is partitioned into nine submatrices, A, B, C, D, E, and F plus three zero submatrice shown as blackened or opaque regions of the W_{ij} mask. The LED array represents the state of the neurons, assumed to be unipolar binary (LED on = neurons firing, LED off = neuron not-firing). The W_{ij} mask represents the strengths of interconnection between neurons in a manner similar to earlier arrangements [11]. Light from the LEDs is smeared vertically over the W_{ij} mask with the aid of an anamorphic lens system (not shown in Fig. 1(a)) and light emerging from rows of the mask is focused with the aid of another anamorphic lens system (also not shown) onto elements of the photodetector (PD) array. Also we assume the same scheme utilized in [11] for realizing bipolar values of W_{ij} in incoherent light is adopted here, namely by separating each row of the W_{ij} mask into two subrows and assigning positive values of W_{ij}^{+} to one subrow and negative values W_{ij}^{-} to the other, then focusing light emerging from the two subrows separately onto pairs of adjacent photosites connected in opposition in each of the V_1, V_2 and H segments of the photodetector array. Submatrix A, with N_1xN_1 elements, provides the interconnection weights of units or neurons within group V_1. Submatrix B, with N_2xN_2 elements, provides the interconnection weights of units within V_2. Submatrices C (of N_1xN_3 elements) and D (of N_3xN_1 elements) provide the interconnection weights between units of V_1 and H and similarly submatrices E (of N_2xN_3 elements) and F (of N_3xn_2) provide the

interconnection weights of units V_2 and H. Units in V_1 and V_2 can not communicate with each other directly because locations of their interconnectivity weights in the W_{ij} matrix or mask are blocked out (blackened lower left and top right portion of W_{ij}). Similarly units within H do not communicate with each other because locations of their interconnectivity weights in the W_{ij} mask are also blocked out (center blackened square of W_{ij}). The LED element θ can be of graded response. It can be viewed as representing the state of an auxiliary neuron in the net that is always on to provide a threshold level to all units by contributing to the light focused onto only negative photosites of the PD arrays by suitable modulation of pixels in the G column of the interconnectivity mask. This method for introducing the threshold level is attractive as it allows for introducing a fixed threshold (fixed θ-LED output) to all neurons or an adaptive threshold if desired. The threshold is global when the transmittances of pixels in G are fixed and the θ LED level in controlled. The threshold is local if the θ LED output is fixed and the pixel transmittances are allowed to vary.

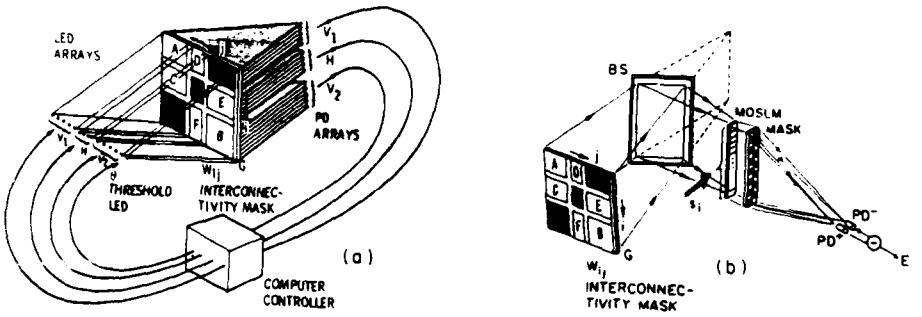


Fig 1. Architecture for opto-electronic analog of layered self-programming net. (a) partitioning concept and, (b) arrangement for rapid determination of the net's global energy E for use in learning by simulated annealing.

We have described elsewhere in some detail [28], how by using a computer controlled nonvolatile spatial light modulator to implement the W_{ij} mask in Fig. 1(a) and including a computer/controller as shown and by repeated application of the simulated annealing procedure with Boltzmann or other stochastic state update rule and collection of co-occurance statistics on the states of the neurons at the end of each run when the net reached thermodynamic equilibrium, the scheme can be made self-programming with ability to modify the weights of synaptic links between its neurons to form internal representations of the input/output associations or patterns presented to it.

3. ACCELERATED LEARNING

The stochastic learning by simulated annealing procedure was originally conceived for serial computation. When dealing with parallel optical computing systems of the kind we described, it does not make sense to strictly adhere to a serial algorithm. Modifications that can take advantage of the available parallelism of optics to speed up stochastic learning should be considered. In this section we offer several such modifications that can markedly speed-up stochastic learning in opto-electronic implementations as compared to serial digital implementation.

Learning by simulated annealing requires calculating the energy E, of the net [7],[29],

$$E = -\frac{1}{2}\sum_{i} u_i s_i \tag{1}$$

where s_i is the state of the i-th neuron and

$$u_i = \sum_{j\neq i} W_{ij} v_j - \theta_i + I_i \tag{2}$$

is the activation potential of the i-th neuron, with θ_i iand I_i being respectively the threshold level and external input of the i-th neuron. Equation (2) can be written in the form,

$$u_i = \sum_{j\neq i} W_{ij} s_j \tag{3}$$

by absorbing θ_i and I_i in the weight matrix W_{ij}. This can be done by adding a G column to the W_{ij} matrix to furnish θ_i as described earlier. A similar procedure can be used to furnish I_i by adding another column with transmittances proportional to I_i whose light transmittance is focused onto the positive photosites of the photodetector array in Fig. 1(a). The above method of introducing θ_i and I_i suggests also that random (noise) components of both θ_i and I_i can be introduced by focusing a random array of light spots, whose intensities are allowed to vary randomly and independently with time, directly onto the positive and negative photosites of the PD array of Fig. 1(a). In this fashion deterministic and random composition of θ_i and I_i can be realized. Taken together, the random components of θ_i and I_i can be viewed as random bipolar noisy threshold. We will return to this point later in our discussion of annealing with noisy threshold.

A simplified version of a rapid scheme for obtaining E opto-electronically is shown in Fig. 1(b). The scheme requires the use of an electronically addressed nonvolatile binary (on-off) spatial light modulator consisting of a single column of N pixels. A suitable candidate is a parallel addressed magneto-optic spatial light modulator (MOSLM) [30], in particular one consisting of a single column of N pixels that are driven electronically by the same signal driving the LED array in order to represent the state vector $\bar{s}$ of the net. A fraction of the focused light emerging from each row of the W_{ij} mask is deflected by the beam splitter BS onto the individual pixels of the column MOSLM such that light from adjacent

pairs of subrows of W_{ij} fall on one pixel of the MOSLM. The MOSLM pixels are overlayed by a checkered binary mask as shown. The opaque and transparent pixels in the checkered mask are staggered in such a fashion that light emerging from the left subcolumn will originate from the positive subrows W_{ij}^{+} of W_{ij} only and light emerging from the right subcolumn will originate from the negative subrows W_{ij}^{-} of W_{ij}. By separately focusing the light from the left land right subcolumns as shown onto two photodetectors and subtracting and halving their outputs one obtains,

$$E = -\frac{1}{2}\sum_{i}[(\sum_{j\neq i} W_{ij}^{+} - W_{ij}^{-})s_j]\, s_i = -\frac{1}{2}\sum_{i}(\sum_{j\neq i} W_{ij}s_j)s_i = -\frac{1}{2}\sum_{i} u_i s_i \qquad (4)$$

which is the required global energy.

Stochastic learning by simulated annealing in the opto-electronic neural net analogs of Fig. 1 requires, as detailed elsewhere [28], fast random number generation for use in random drawing and switching of state of neurons from H and from H and V_2. Another random number is also needed to execute the stochastic state update rule. Although fast digital pseudo-random number generation of up to 10^9 [sec^{-1}] is feasible [31] and can be used to help speed up digital simulation of the learning algorithm, this by itself is not sufficient to make a large impact specially when the total number of neurons in the net is large. Opto-electronic random number generation is also possible although at a slower rate of about 10^5 [sec]. Despite the slower rate of generation, opto-electronic methods have advantages that will be elaborated upon below. An opto-electronic method for generating the Boltzmann probability factor needed in the simulated annealing algorithm [28] employing speckle statistics is described in [32] and optical generation of random number arrays by photon counting image acquisition systems or clipped laser speckle have also been recently described [33]-[36]. These photon counting image acquisition systems have the advantage of being able to generate normalized random numbers with any probability density function. A more important advantage of optical generation of random number arrays however is the ability to exploit the parallelism of optics to modify the simulated annealing and the Boltzmann machine formalism detailed above in order to achieve significant improvement in speed. As stated earlier, with parallel optical random number generation, a spatially and temporally uncorrelated linear array of perculating light spots of suitable size can be generated and imaged onto the photodetector array (PDA) of Fig. 1 directly such that both the positive and negative photosites of the PDA are subjected to random irradiance. This introduces a random (noise) component in θ_i and I_i of eq. (2) which can be viewed as stated earlier as bipolar noisy threshold. The noisy threshold produces in turn a noisy component in the energy in accordance to eq. (2). The magnitude of the noise components can be controlled by varying the standard deviation of the random light intensity array irradiating the PDA. The noisy threshold produces therefore random controlled perturbation or "shaking" of the energy landscape of the net. This helps shake the net loose whenever it gets trapped in a local energy minimum. The procedure can be viewed as generating a controlled deformations or tremor in the energy landscape of

the net to prevent entrapment in a local energy minimum and ensure thereby convergence to a state of global energy minimum. Both the random drawing of neurons (more than one at a time is now possible) and the stochastic state update of the net are now done in parallel at the same time. This leads to significant acceleration of the simulated annealing process. In the following, results of numerical simulations aimed at gaining insight in the performance of the noisy threshold scheme are presented.

4. SIMULATION RESULTS

For the purposes of this simulation we form a fully interconnected (single layer) neural net with random bipolar binary weights matrix with diagonal elements set to zero. The number of neurons N is 16. The weights are symmetrical. Figure 2 shows the density of states (energy histogram) of the net, where we calculate the energies of all the possible (2^{16}) configurations of the net. The Y (vertical) axis shows the number of configurations (out of 2^{16}) with the same energy and their corresponding energy energy represented by the X (horizontal) axis. The low energy configurations correspond to states near the very left of the curve, of which a good annealing scheme should find one. To avoid lengthy simulation time, we do not in the following exhaust the simulation for all possible configurations (2^{16}). Instead, we randomly select 50 configurations as the test sample space. The energy histogram of these 50 configurations is shown in Fig. 3. In Fig. 4 is shown the energy histogram of the states to which the net converges when initiated with the 50 configurations of the test space. The histogram was obtained by initiating the net with any one of the 50 states and followed by finding the final state to which net net converges by iteratively applying the customary neural net state update rule [7] namely,

$$s_i = \begin{cases} 1 & \text{if } u_i > 0 \\ 0 & \text{if } u_i \leq 0 \end{cases} \tag{4}$$

which amounts to performing a steepest gradient descent search into local minima of the net, then calculating the energy of the final state. The plot means that there are Y number of initial states (out of the 50 configurations of the sample space) which converge (in the sense of the above conventional steepest descent) to local minima with same energy X. We see that a fair number of initial states end up trapped in local energy minima at high energy state because the steepest gradient descent search method involved is deterministic and does not have provisions for escaping from a local minimum. This curve can serve as a reference to test the performance of an annealing scheme's ability to escape from a local minimum and find the global minimum. In Figs 5 and 6, we display the energy histograms of the convergent states when different annealing schemes were used. In these figures, each of the 50 configurations of test space is used as input vector for 100 times, and the statistics of the convergent states are collected. The results employing the simulated annealing algorithm [23],[24] with random drawing of neurons one neuron at a time, and

stochastic update employing noise uniformly distributed in the range (-T,T) are shown in Fig. 5. The annealing or cooling schedule used was: 96 @ 10, 160 @ 5, 96 @ 3, 96 @ 1, 96 @ .5, and 96 @ 0.01 where I @ T specifies the number of iterations I at temperature T. Figure 6 shows the results obtained with the noisy threshold scheme where the deterministic component of the threshold is taken to be zero and independent bipolar noise components uniformly distributed in the (-T,T) range are added to the thresholds every iteration. The probability of the i-th neuron switching its state was taken to be inversely proportional to its activation potential u_i. The noise amplitude T was reduced gradually every specified number of iterations to allow the net to find the states of global energy minimum or one close to it. The following annealing schedule was utilized: 10 @ 2.5, 10 @ 1.5, 10 @ .5, and 10 @ .1. It is seen that annealing with noisy threshold finds states of global energy minima equally well as the conventional simulated annealing scheme. The number of iterations involved is however considerably less: 40 as compared to 540 in the conventional simulated annealing scheme. It is worth noting also that the noisy threshold scheme does not require knowing the energy of the net to apply the rule described in the preceeding section. This further accelerates the search for the global minimum and can markedly shorten learning time [28].

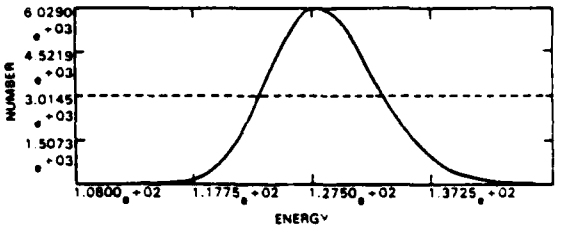


Fig. 2. Density of states for 2^{16} possible configurations.

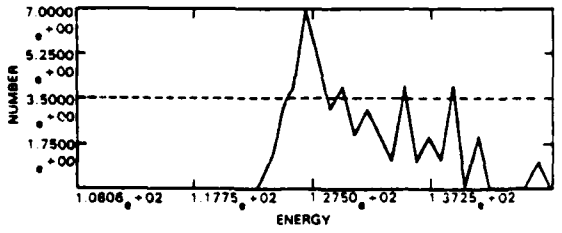


Fig. 3. Density of states of the sample space (of 50 randomly selected configurations).

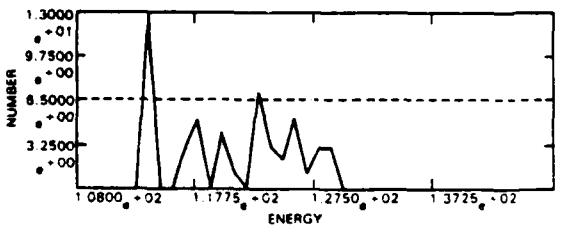


Fig. 4. Density of convergent states in sample space (conventional neural net update rule).

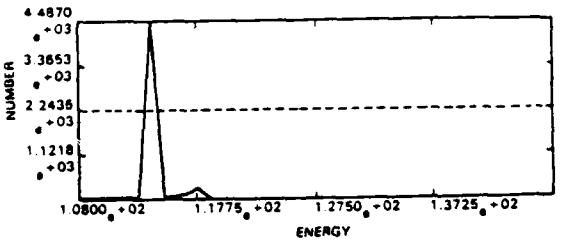


Fig. 5. Density of convergent states in sample space (conventional simulated annealing with noise uniformly distributed in (-T,T)).

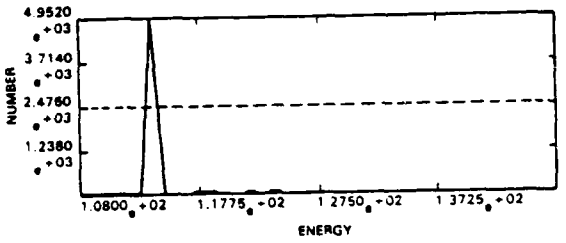


Fig. 6. Density of convergent states in sample space (noisy threshold scheme using noise uniformly distributed in (-T,T).

5. DISCUSSION

We have described an architecture for partitioning an opto-electronic analog of a neural net to form a multilayered net that permits self-organization and learning when computer controlled nonvolatile spatial light modulators are utilized to realize the required plasticity. The focus here is on stochastic learning as opposed to deterministic learning because it can account for the role of noise in biological neural nets. We also described opto-electronic architectures that can be used for fast determination of the energy of the net and therefore can accelerate the simulated annealing process involved in stochastic learning where "optical random arrays" can also be used to accellerate the process further. However, when parallel optical computing is employed, it is not necessary to adhere to a serial simulated annealing algorithm. We have shown that departure from the conventional simulated annealing algorithm through the use of a noisy thresholding scheme promises to markedly accelerate stochastic learning in opto-electronic implementation of multilayered neural nets, make the procedure more biologically plausible, and make stochastic learning practical.

6. ACKNOWLEDGEMENT

The work reported was supported by grants from DARPA/NRL, The Army research Office, and the University of Pennsylvania Laboratory for Research on the Structure of Matter. Some of the work reported was performed when one of the authors (NF) served as distinguished visiting scientist at NASA-JPL.

REFERENCES

1. Widrow, B. and M.E. Hoff, WESCON Convention Record, Part IV, pp. 96-104, 1960. Also in Self Organizing Systems, M.C. Yovitz, et. al. (Eds.) Spartan 1962.
2. Nakano, K., IEEE Trans. on Systems Man & Cybernetics, SMC-2, 380, 1972.
3. Kohonen, T., IEEE Trans. on Computers, C-21, 353, 1972.
4. Kohonen, T., Associative Memory, Springer Verlag, Heidelberg (1978). Also "Self-Organization and Associative Memory", Springer Verlag, 1984.
5. Willshow, D.J., Proc. Roy. Soc. London, 182, 233, 1972.
6. Anderson, J.A., et. al., Physiological Review, 34, 413, 1977.
7. Hopfield, J.J., Proc. Natl. Acad. Sci., USA, 79, 1982. Also, Proc. Natl. Acad. Sci., USA, 81, 1984.
8. Grossberg, S., Studies of Mind and Brain, R. Reidel Pub. Co., Boston, 1982.
9. Sanderson, A.C., and Y.Y. Zeevi, (Eds.), Special Issue on Neural and Sensory Information Processing, IEEE Trans. on Syst. Man and Cybernetics, SMC-13, 1983.
10. Psaltis, D. and N. Farhat, Opt. Lett., 10, 98, 1985.
11. Farhat, N.H., et. al., App. Optics, 24, 1460, 1985.
12. Farhat, N.H., and D. Psaltis, Digest OSA Annual Meeting, Wash., D.C., p. 58, 1985.
13. Lee, K.S. and N.H. Farhat, Digest OSA Annual Meeting, Wash., D.C., p. 48, 1985.
14. Psaltis, D., et. at., Digest OSA Annual Meeting, Wash. D.C., p. 58, 1985.
15. Condon, D.J., et. al., OSA Topical Meeting on Optical Computing, Incline Village, Nov. 1985, (Post deadline paper).
16. Fisher, A.D., C.L. Giles, and J.N. Lee, J. Opt. Soc. Am. 1, 1337, 1984.
17. Fisher, A.D., and C.L. Giles, Proc. of the IEEE COPCOM Spring Meeting, IEEE Computer Society, IEEE Cat. No. CH2135-2/85/0000/0342, pp. 342, 1985.
18. Fisher, A.D., et. a., SPIE, 625, Bellingham, WA., 1986.
19. Sofer, B., et. al., Opt. Lett., 11, 118, 1986.
20. Anderson, D.Z., Opt. Lett. 11, 56, 1986.
21. Yariv, A. and S.K. Kwong, Opt. Lett., 11, 186, 1986.
22. Farhat, N., et. a., in Neural Networks for Computing, AIP Conf. Proc. 151, J.S. Denker Editor, 146, (1986).
23. Metropolis, N., et. al., J. Chem. Phys., 21, 1087, 1953.
24. Kirkpatric, S., et. al., Science, 220, 671, 1983.
25. Hinton, G.F., et. al., Carnegie-Mellon University Report No. CMU-CS-84-119, May (1984).
26. Sejnowski, T.J., and C.R. Rosenberg, Johns Hopkins University, Electrical Engineering and Computer Science Technical Report No. JHU/EECS-96/01 (1986).
27. Rumelhart, D.E., G.E. into and R.J. Williams, Institute for Cognitive Science Report 8506, Univ. of California, San Diego, Sept. 1985.
28. Farhat, N., Optics Letters, (accepted for publication).
29. Cohen, M.A., and S. Grossberg, Trans. on Man Syst. and Cybern., SMC-13, 815, 1983.
30. Ross, W., D. Psaltis, and R. Anderson, Optical Engineering, 22, 485, 1983.

31. Kirckpatric, S. Private Communication (1987).
32. Ticknor, A.J., H.H. Barret and R.L. Easton, Jr., Post-Deadline Paper PD-3, OSA Topical Meeting on Optical Computing, Incline Village, Nev. (1985).
33. Morris, G.M., Optical Engineering, 24, 86, (1985).
34. Marron, J., A. Martina and G.M. Morris, App. Optics, 25, 26 (1986).
35. Devos, F., P. Garda, and P. Chavel, Optics Letters, 12, 152, (1987).
36. Tsuchiya, Y., E. Inuzuka, T. Kurono, and M. Hosada, in Advances in Electron Devices, B.L. Morgan (Eds.), 21, Academic Press (1985).

Bimodal Stochastic Optical Learning Machine

N. H. Farhat Z. Y. Shae

University of Pennsylvania
Electrical Engineering Department
Phila. PA 19104

Abstract

Self-organization and learning is an important attribute of neural nets that sets them apart from other approaches to signal processing. The study of stochastic learning by simulated annealing in the context of a Boltzmann machine is attractive because it could shed light on the role of noise in biological neural nets and because it can lead to artificial neural nets that can be switched between two distinct operating modes depending on noise level in the network. At finite noise level (or temperature) the net can be operated in a "soft" mode where learning can take place by automated synaptic modifications. Once learning is completed the net is "hardened" (or frozen) and acts as associative memory by reducing the noise level or temperature to zero. We present the results of numerical and experimental study aimed at opto-electronic realization of such networks. The results include: (a) fast annealing by noisy thresholding which demonstrates that the global energy minimum of a small analog test network can be reached in a matter of a few tens of neuron time constants, (b) stochastic learning with binary weights which paves the way for the use of fast binary and nonvolatile spatial light modulators to realize synaptic modifications.

1 System Architecture

Optics and opto-electronic architectures and techniques can play an important role in the study and implementation of self-programming networks and in speeding-up the execution of learning algorithms. Learning requires partitioning a net into layers with a prescribed communication pattern among them. A method for partitioning an opto-electronic analog of a neural net into input, output, and internal groups (layers) of neurons with selective communication pattern among neurons within each layer and between layers that is capable of stochastic learning, by means of a simulated annealing algorithm in the context of a Boltzmann machine formalism is described in Fig. 1(a) The network, consisting of N neurons, is partitioned into three groups. Two groups, V_1 and V_2, represent visible or environmental units that can be used as input and output units respectively. The third group H are hidden units. The partition is such that $N_1 + N_2 + N_3 = N$ where subscripts 1,2, and 3 on N refer to the number of neurons in the V_1, V_2 and H groups respectively. The interconnectivity matrix, designated

here as w_{ij}, is partitioned into nine submatrices, A, B, C, D, E, and F plus three zero submatrices shown as blackened or opaque regions of the w_{ij} mask. The LED array represents the state of the neurons, assumed to be unipolar binary (LED on = neurons firing, LED off = neurons not-firing). The w_{ij} mask represents the strengths of the interconnection between neurons. Light from the LEDs is smeared vertically over the w_{ij} mask with the aid of an anamorphic lens system (not shown in Fig. 1(a)) and light emerging from rows of the mask is focused with the aid of another anamorphic lens system (also not shown) onto elements of the photodetector (PD) array. Bipolar values of w_{ij} can be realized in incoherent light by separating each row of the w_{ij} mask into two subrows and assigning positive values of w_{ij}^{+} to one subrow and negative values w_{ij}^{-} to the other, then focusing light emerging from the two subrows separately onto pairs of adjacent photosites connected in opposition in each of the V_1, V_2 and H segments of the PD array as described elsewhere [2]. Submatrix A, with N_1xN_1 elements, provides the interconnection weights of units or neurons within group V_1. Submatrix B, with N_2xN_2 elements, provides the interconnection weights of units within V_2. Submatrices C (of N_1xN_3 elements) and D (of N_3xN_1 elements) provide the interconnection weights between units of V_1 and H and similarly submatrices E (of N_2xN_3 elements) and F (of N_3xN_2 elements) provide the interconnection weights of units V_2 and H. Units in V_1 and V_2 can not communicate with each other directly because locations of their interconnectivity weights in the w_{ij} matrix or mask are blocked out (blackened lower left and top right portion of w_{ij}). Similarly units within H do not communicate with each other because locations of their interconnectivity weights in the w_{ij} mask are also blocked out (center blackened square of w_{ij}). The LED element θ is of graded response. It can be viewed as representing the state of an auxiliary neuron in the net that is always on to provide a threshold level to all units by contributing to the light focused onto only negative photosites of the PD array by suitable modulation of pixels in the G column of the interconnectivity mask. This method for introducing the threshold level is attractive as it allows for introducing a fixed threshold to all neurons or an adaptive threshold if desired. It can also be employed to alter the energy landscape of the net adaptively in accordance to the behavior of other parameters of the net. Figure 1(b) shows the arrangement for rapid determination of the net's energy E for use in learning by simulated annealing. A computer works as the system controller to calculates P_{ij} and P'_{ij}, and also to control the MOSLM which implements the interconnectivity matrix W. This architecture allows stochastic learning by simulated annealing in the context of a Boltzmann machine. The learning algorithm for Boltzmann machine can be summarized as follows:

1. Choose one mapping or associated pair that the net is required to learn, and present it to the net. The associated pair consists of two unipolar binary vectors one an input vector and the other an output vector.

2. Clamp the input vector to the V_1 neurons, and the corresponding output vector to the V_2 neurons.

3. Employ simulated annealing method in energy space to find low energy configurations at the given V_1 and V_2. The final temperature in the cooling schedule is

called T_0 and will be used later as an annealing parameter in Cross-Entropy or G-space. During this step, random drawing and change of only the states of the hidden neurons (H) takes place.

4. Repeat steps 2-3 V1 times for all associations the net is required to learn, and collect co-occurrence statistics i.e. determine the probabilities P_{ij} of the ith and jth being in the same state i.e. both being on or off.

5. Unclamp the V_2 neurons and repeat steps 3-4 for all input vectors, and collect co-occurrence statistics again i.e. determine the probabilities P'_{ij} of the ith and jth neurons being in the same state. During this step, random drawing and change of both the states of the H and the V_2 neurons takes place.

6. All weights in the net are modified by increasing the synaptic weight (W_{ij}) between the ith and jth neurons by a small amount δ if $P_{ij} - P'_{ij} > 0$, otherwise, decreasing the weight by the same amount. Note this requires multivalued W_{ij} or incremental variation of W_{ij} that requires the use of graded response spatial light modulators for realizing synaptic modifications in opto-electronic implementations.

7. We call steps 1-6 a learning cycle. The learning cycle consists of two phases. Phase one involves clamping the input and output units to the associated pairs. Phase two involves clamping the input units to the input vector alone and letting the output units free run with the hidden units. The learning cycle is repeated again and again and is halted after $P_{ij} - P'_{ij}$ is close to zero for every i and j.

The learning procedure described above can be supported in the opto-electronic hardware environment described previously.

2 Fast Annealing With Noisy Threshold

With the aid of an optical random number generation, a spatially and temporally uncorrelated linear array of perculating light spots of suitable size and intensity range can be generated and imaged onto the PD array of Fig. 1 directly such that both the positive and negative photosites of the PD array are subjected to random irradiance. This introduces a random (noise) component in the threshold. The noisy threshold produces in turn a noisy component in the energy function of the net. The magnitude of the noise components can be controlled by varying the light intensity array irradiating the PD array. The noisy threshold produces therefore random controlled perturbation or "shaking" of the energy landscape of the net. This helps shake the net loose whenever it gets trapped in a local energy minimum. The procedure can be viewed as generating a controlled gradually decreasing deformations or tremors in the energy landscape of the net that prevents entrapment in a local energy minimum and helps the net settle into the global minimum energy state or one close to it. Both the random drawing of neurons (more than one at a time is now possible) and the stochastic state update of the net are now done in parallel at the same time. This leads to significant acceleration

of the simulated annealing process. Electronic control of the random light intensity enables realizing any annealing profile. We had presented the results of numerical study elsewhere [1]. In the following, results of an experimental study aimed at gaining insight in the performance of the noisy threshold scheme are presented.

3 Experimental Results

An annealing experiment based on the noisy threshold algorithm in an opto-electronic neural net is reported. A television screen tuned to an empty channel where no TV station operates is used as the spatio-temporal optical noise source. We use a lens to project the optical noise pattern (snow pattern) onto the photodetector array of an opto-electronic neural net consisting of 16 unipolar binary neurons of the type described elsewhere [2]. The connectivity matrix of the network was the same random ternary matrix utilized in earlier work [1]. The brightness of the TV screen is controlled by the D/A output of a MASSCOMP computer, and the convergent state is monitored by the A/D input of the same computer. A photograph of the experimental arrangement is shown in Figure 2. We investigated four types of cooling profiles: linear, concave, convex, and stair-case illustrated in Figure 3. For each cooling profile, we investigate 5 annealing time intervals: 100, 200, 500, 1000, and 2000 ms. For each cooling profile and annealing time interval, we do the annealing 100 times to collect sufficient statistics, and find the probability that the system converges to its global minimal energy state. The experimental results obtained show that the setup can find the global energy minimum of an artificial neural net of 16 neurons in 2000 ms which corresponds to 32 time constants of the neurons in the test network. A net of neurons with response time of 1 μ sec would anneal therefore in few tens of microseconds and this is expected to be independent of the number of neurons in the net as long as parallel injection of noise in the network is implemented. The cooling profile had no observable effect on this result. The probabilities of convergence to a global minimum as function of the annealing duration for different annealing profiles are shown in the table 1.

4 Stochastic Learning With Binary Weights

The Boltzmann machine learning algorithm described earlier employs graded weights. However, from practical viewpoint, learning in artificial neural nets can be simplified considerably if binary weights can be used. This would pave the way to using fast nonvolatile binary spatial light modulators (SLMs) such as Magneto-Optic SLM and Ferroelectric liquid crystal SLM. However, a Boltzmann machine basically is an adaptive system. If the step size of adaptive changes is too large and the sensitivity of system response to the error signal is high, the machine will generally become unstable. Since a traditional Boltzmann machine has a high sensitivity in response to error signal, i.e., it responds to the error signal $(P_{ij} - P'_{ij})$ to modify synaptic weights even when the error signal is very small, small weight variations are required to prevent the system from becoming unstable. However, in a binary weight net $(W_{ij} = 1, -1)$ the step size of adaptive change is large and fixed (-2 or 2). In order to prevent the

system from becoming unstable, we increase the inertia of weights i.e. weights do not change when small value of $P_{ij} - P'_{ij}$ occurs. As a result, the learning procedure of the Boltzmann machine in a binary weight net would be identical to the procedure of the graded weights net stated in the system architecture section, except step 6 which is modified as follows: If $P_{ij} - P'_{ij} \geq M$, set $W_{ij} = 1$; if $P_{ij} - P'_{ij} \leq -M$, set $W_{ij} = -1$; otherwise, no change, where $M \in [0, 1]$ is a fixed constant.

The goal of the Boltzmann machine is to minimize the Cross-Entropy G by means of modifying the weights of the net in a certain order. The G space is an information theoretic measure of the distance between the probability distributions when an environmental input is present in the net and when it is free running with no environmental input applied, and is given by

$$G = \sum_{\alpha} P^{+}(V_\alpha) \ln \frac{P^{+}(V_\alpha)}{P^{-}(V_\alpha)} \tag{1}$$

where $P^{+}(V_\alpha)$ is the probability of the visible units being in the α state when the visible units are subjected to the environmental input. Namely, $P^{+}(V_\alpha)$ represents the desired or specified probability for the α state. $P^{-}(V_\alpha)$ is the corresponding probability when the net is free-running. Namely, $P^{-}(V_\alpha)$ represents the actual probability generated from the net for the α state. $P^{-}(V_\alpha)$ depends on the weights W_{ij}, and so G can be altered by changing W_{ij}. Since, in general, there are local minima in G space, gradient descent search will find a local minimum instead of the global minimum. In order to reach the global minimum in G space, introduction of noise in G space is required. However, if the noise level is too large, the network can not generate the specified or desired environmental distribution. A systematic way for adding noise in G space, i.e. an annealing scheme in G space, has not yet been studied in detail. Here we propose the use of the final temperature T_0 of the simulated annealing schedule used in the energy space E as the annealing parameter in G space, since $P^{-}(V_\alpha)$ is function of T_0. In the first few learning cycles, we use high values of T_0. This will provide high level of noise in G space. The value of T_0 is decreased gradually along with the number of learning cycles. Accordingly, a simulated annealing process in G space is accomplished by decreasing the final temperature T_0 in a similar way to the simulated annealing process in energy space which is accomplished by decreasing annealing temperature T. Note that an annealing schedule with high value of T_0 is equivalent to a short time interval annealing schedule in E space, i.e., both cases can generate high level of noise in G space, and vice versa. Accordingly, annealing time interval in E space can also be used as an annealing parameter in G space. As a result, a simulated annealing process in G space can also be accomplished by gradually increasing the annealing time interval in E space along with the number of learning cycles. Results of computer simulations of stochastic learning by simulated annealing in a Boltzmann machine employing both graded and binary weights are presented in the next section.

5 Simulation Results

In these simulations we use noisy threshold (N-T) annealing scheme [1] and use the annealing time interval in E space as an annealing parameter in G space. All the simu-

lations learn to solve a 4-2-4 encoder problem [3] in the context of Boltzmann machine formalism i.e. this consists of having a three layered net, of the kind described in the architecture section, learn to form its own internal representations of the associations presented to it. For all simulations, the net reaches equilibrium 100 times (25 times for each input vector) for collecting the statistics of P_{ij} during the input and output clamping phase. The situation is the same for collecting the statistics of P'_{ij}. All annealing schedules are stated in the corresponding Figures in the notation of $I@T$ explained earlier [1]. The noise we used is binary noise whose amplitude is either T or $-T$ and is decreased gradually in time and terminated at at T_0. Figure 4 shows the results of the linear weight learning scheme, and Figures 5 shows the results of the binary weight learning scheme. All Figures show the results for 12 runs. The parameter M we used is 0.1. Only two annealing schedules in E space are used for the annealing in G space. During the first half of the total number of learning cycles the short time interval annealing schedule is employed, and during the later half of the learning cycles the long time interval annealing schedule is employed. These results show the viability of the annealing scheme in G space, and also show the viability of the binary weight stochastic learning scheme.

6 Conclusions

We have described an architecture for partitioning an opto-electronic analog of a neural net to form a multilayered net that permits self-organization and learning when computer controlled nonvolatile spatial light modulators are utilized to realize the required plasticity. The focus here is on stochastic learning as opposed to deterministic learning because it may provide insight in the role of noise in biological neural nets. We also described opto-electronic architectures that can be used for fast determination of the energy of the net if such information is needed and for adaptive deterministic deformation of the net's energy landscape to control its behavior. We show that departure from the conventional simulated annealing algorithm through the use of noisy thresholding in opto-electronic schemes promises to markedly accelerate the annealing process, and make stochastic learning practical. Employing the noisy thresholding scheme a small opto-electronic neural net (of 16 neurons) was found to reach a global energy minimum or one close to it in about 32 neuron time constants. We also show that binary weight learning algorithm can be used in the context of a modified Boltzmann machine. This paves the way to the use of nonvolatile binary spatial light modulators to realize the required plasticity in such stochastic learning nets. Such nets, having learned their environmental inputs can be "frozen" for use as associative memories of the entities learned by merely removing injected noise from the net. Noise injection for annealing returns the nets to a "soft" mode for learning new environmental inputs.

7 Acknowledgement

This research was supported by grants from DARPA and the U. S. Army Harry Diamond Laboratory.

8 References

1. Nabil H. Farhat and Z. Y. Shae, "Architectures and Methodologies for Self-Organization and Stochastic Learning in Opto-Electronic Analogs of Neural Nets", IEEE first annual conference on neural net, ICNN, San Diego, California, June (1987)

2. Nabil H. Farhat, Demetri Psaltis, Aluizio Prata, and Eung Paek, "Optical implementation of the Hopfield model", Applied Optics, Vol. 24, pp. 1469-1475, May 15, (1985)

3. G. E. Hinton, T. J. Sejnowski, and D. H. Ackley, "Boltzmann machines: Constraint Satisfaction Networks That Learn", Carnegie-Mellon University, Technical Reports, CMU-CS-84-119, (1984)

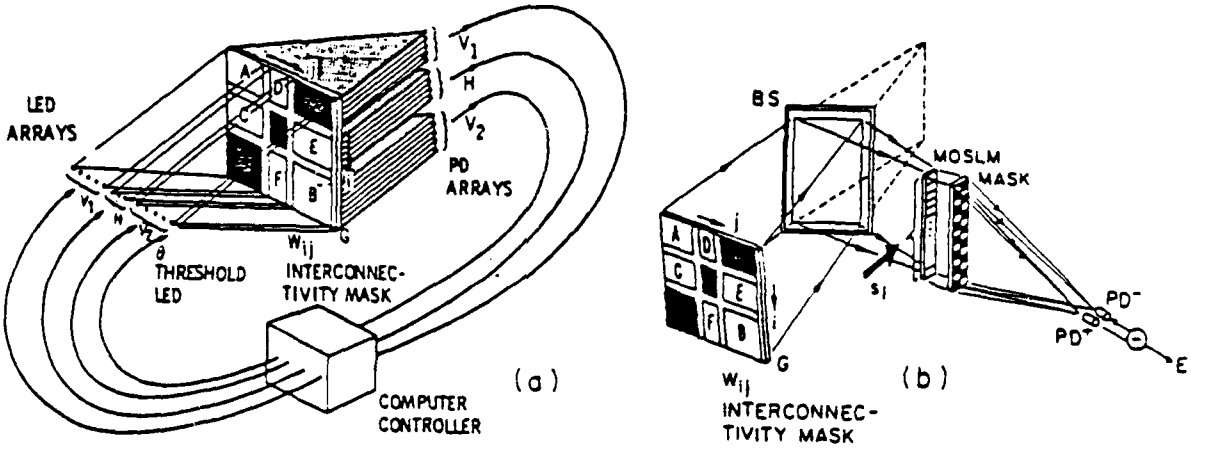


Fig. 1. Architecture for opto-electronic analog of layered self-programming net. (a) partitioning concept and, (b) arrangement for rapid determination of the net's global energy E for use in learning by simulated annealing.

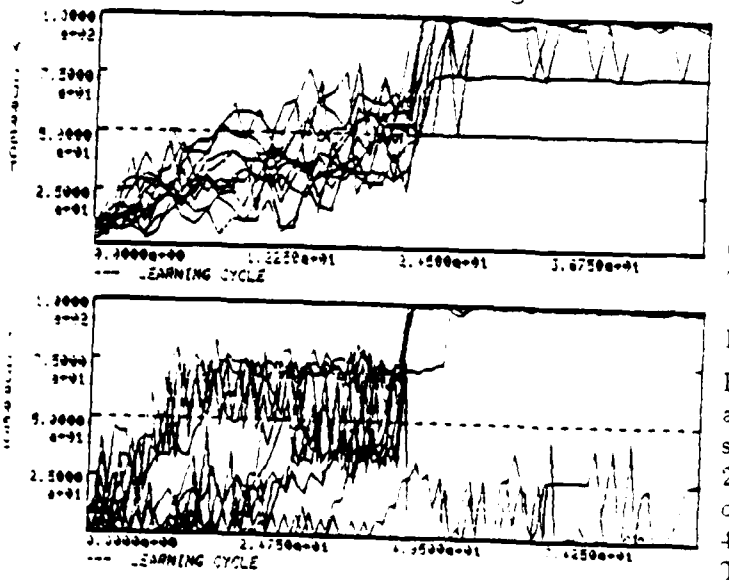


Fig. 4.

Linear weight learning curve with N-T algorithm. Annealing schedule in E space: during the 0-25th learning cycle 2 @ 3, 1 @ 1.5, 1 @ 1, and 2 @ 0.1: during the 26-50th learning cycle 6 @ 1, 6 @ 0.8, 6 @ 0.5, 6 @ 0.1, and 6 @ 0. This is a annealing scheme in G space.

Fig. 5.

Binary weight learning curve with N-T algorithm. Annealing schedule in E space: during the 0-50th learning cycle 2 @ 3, 1 @ 1.5, 1 @ 1, and 2 @ 0.1: during the 51-100th learning cycle 4 @ 1, 4 @ 0.8, 4 @ 0.5, 4 @ 0.1, and 4 @ 0. This is a annealing scheme in G space.

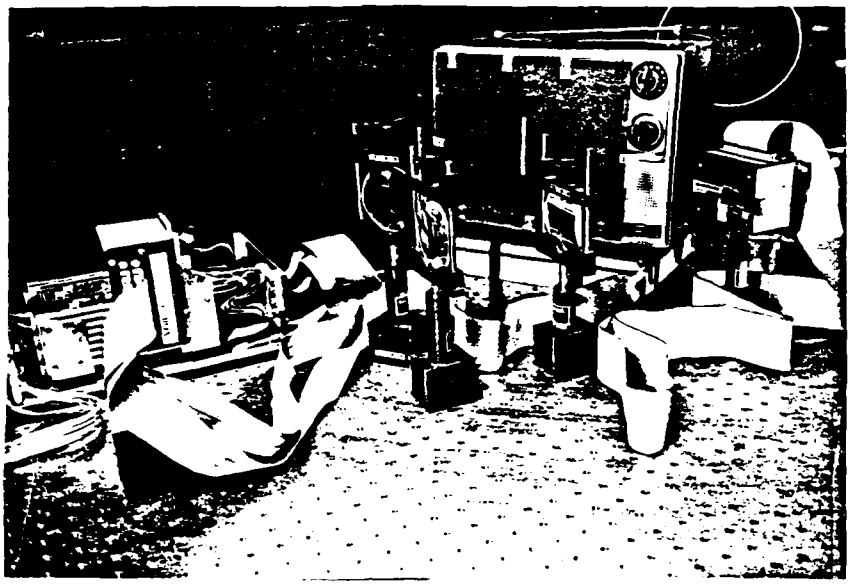


Fig. 2. Pictorial view of opto-electronic neural net of 16 unipolar binary neurons with random ternary weights used to verify fast annealing by noisy threshoolding.

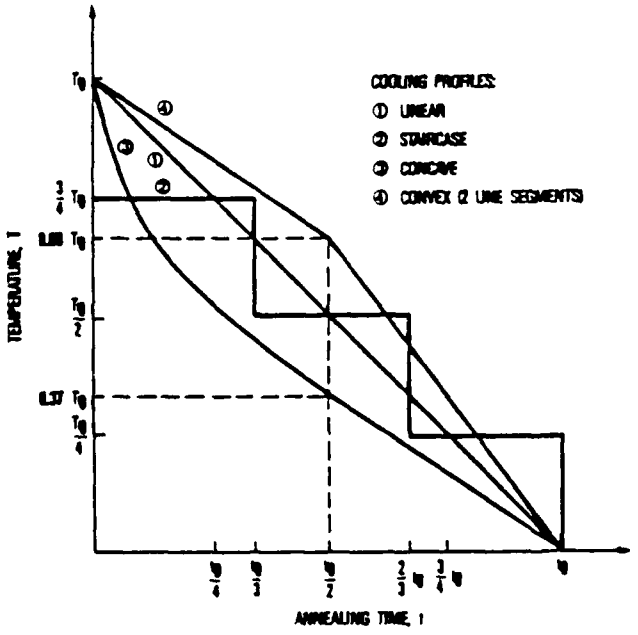


Fig. 3. Cooling profiles

Duration \ Profile	*linear*	*concave*	*convex*	*step*
$100ms$	0.48	0.46	0.50	0.45
$200ms$	0.62	0.56	0.68	0.64
$500ms$	0.78	0.73	0.77	0.79
$1000ms$	0.83	0.86	0.84	0.88
$2000ms$	0.97	0.96	0.96	0.98

Table 1 The probabilities of convergence to a global minimum as function of the annealing duration for different annealing profiles.